

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 generics

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

Properties are not strict for all instances, because

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

Assignment Project Exam Help

<https://powcoder.com>

Generics vs. Universals

4 Violins have four strings

vs.

5 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 in general, but not necessarily

reasonable to conclude $Q(a)$ given $P(a)$
unless there is a good reason not to

Here: qualitative version (no numbers)

Varieties of defaults

General statements

- statistical: Most P 's are Q 's.
People living in Quebec speak French.
- normal: All normal P 's are Q 's.
Polar bears are white.
- prototypical: The prototypical P is a Q .
Owls hunt at night.

Representational

- conversational: Unless I tell you otherwise, a P is a Q .
 - default slot values in frames
 - disjointness in IS-A hierarchy (sometimes)
 - closed-world assumption (below)

Epistemic rationales

- familiarity: If a P was not a Q , you would know it.
 - an older brother
 - very unusual individual, situation or event
- group confidence: All known P 's are Q 's.
NP-hard problems unsolvable in poly time.

Persistence rationale

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

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Closed world assumption

Reiter's observation:

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

AirLine Example: flight guide provides

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

but not: \neg 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

$Negs = \{\neg p \mid p \text{ ground atomic and } KB \not\models p\}$

Note: relation to negation as failure

a common pattern:
 $KB' = KB \cup \Delta$

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

Properties of CWA

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

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 complete knowledge

never any unknowns, relative to vocabulary

In general, a KB has incomplete knowledge,

e.g., if $KB = (p \vee q)$, then $KB \models (p \vee q)$, but
 $KB \not\models p$, $KB \not\models \neg p$, $KB \not\models q$, and $KB \not\models \neg q$

But with CWA, always have:

If $KB \models_c (\alpha \vee \beta)$, then $KB \models_c \alpha$ or $KB \models_c \beta$.
similar argument to above

KR & R © Brachman & Levesque 2005

Defaults

<https://powcoder.com>

Add WeChat powcoder

Query evaluation

With CWA can reduce queries (without quantifiers) recursively to 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: $KB \models_c \lambda$, where λ 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 atomic wffs stored as a table, deciding whether or not $KB \models_c \alpha$ 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

see "vivid reasoning" (discussed later)

KR & R © Brachman & Levesque 2005

Defaults

Consistency

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

Also works if KB has conjunctions and if KB has -ve 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. $KB = (p \vee q)$ $Negs = \{\neg p, \neg q\}$

so $KB \cup Negs$ is inconsistent

and for every α , $KB \models_c \alpha$!

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

One approach: GCWA

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

When KB is consistent, get:

$KB \cup Negs$ consistent

– everything derivable is also derivable by CWA

KR & R © Brachman & Levesque 2005

Defaults

Assignment Project Exam Help

<https://powcoder.com>

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 term t , we have $KB \models_c \neg \text{DirectConnect}(\text{myHome}, t)$ does not mean that $KB \models_c \forall x [\neg \text{DirectConnect}(\text{myHome}, x)]$

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

Define: $KB \models_{c2} \alpha$ iff $KB \cup Negs \cup Dc \cup Un \models \alpha$

$Negs$ is as before

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

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

where the c_i are all the constants appearing in KB (assumed finite)

Get: $KB \models_{c2} \exists x. \alpha$ iff $KB \models_{c2} \alpha[x/c]$,

for some c appearing in the KB

$KB \models_{c2} \forall x. \alpha$ iff $KB \models_{c2} \alpha[x/c]$,

for all c appearing in the KB

$KB \models_{c2} (c = d)$ iff c and d are the same constant

KR & R © Brachman & Levesque 2005

Defaults

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 non-monotonic reasoning

- start with explicit beliefs
- generate implicit beliefs non-monotonically, taking defaults into account
e.g. Birds fly.
- implicit beliefs may not be uniquely determined
vs. monotonic case: $\{\alpha \mid KB \models \alpha\}$

Will consider three approaches:

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

KR & R © Brachman & Levesque 2005

Defaults

Assignment Project Exam Help

<https://powcoder.com>

Minimizing abnormality

Idea:

CWA makes the extension of all predicates as small as possible

by adding negated literals

Generalize: make extension of selected predicates as small as possible

Ab predicates used to talk about defaults

Example:

$\forall x[\text{Bird}(x) \wedge \neg \text{Ab}(x) \supset \text{Fly}(x)]$

All birds that are normal fly

$\text{Bird}(\text{chilly}), \neg \text{Fly}(\text{chilly}), \text{Bird}(\text{tweety}), (\text{chilly} \neq \text{tweety})$

Would like $\text{Fly}(\text{tweety})$, but $KB \not\models \text{Fly}(\text{tweety})$

because there is an interp I where

$\Phi(\text{tweety}) \in \Phi(\text{Ab})$

Solution: consider only interps where $\Phi(\text{Ab})$ is as small as possible, relative to KB

for example: need $\Phi(\text{chilly}) \in \Phi(\text{Ab})$

Generalizes to many Ab_i predicates

KR & R © Brachman & Levesque 2005

Defaults

Minimal Entailment

Given two interps over the same domain, I_1 and I_2

$I_1 \leq I_2$ iff $\Phi_1(Ab) \subseteq \Phi_2(Ab)$
for every Ab predicate

$I_1 < I_2$ iff $I_1 \leq I_2$ but not $I_2 \leq I_1$

read: I_1 is more normal than I_2

Define a new version of entailment, \models_m , by

$KB \models_m \alpha$ iff for every I ,
if $I \models KB$ and no $I^* < I$ s.t. $I^* \models KB$
then $I \models \alpha$.

So only requiring α to be true in interpretations
satisfying KB that are minimal in abnormalities

Get: $KB \models_m \text{Fly}(\text{tweety})$

because if interp satisfies KB and is minimal,
only $\Phi(\text{chilly})$ will be in $\Phi(Ab)$.

Minimization need not produce a unique interpretation:

$\text{Bird}(a), \text{Bird}(b), [\neg \text{Fly}(a) \vee \neg \text{Fly}(b)]$

Two minimal interpretations

$KB \models_m \text{Fly}(a), KB \models_m \text{Fly}(b), KB \models_m [\text{Fly}(a) \wedge \text{Fly}(b)]$

Different from the CWA: no inconsistency!

KR & R © Brachman & Levesque 2005

Defaults

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Circumscription

Can achieve similar effects by leaving entailment
alone, but adding a set of sentences to the KB

like CWA, but not as simple as adding $\neg Ab(t)$ since we
need not have constant names for abnormal individuals

Idea: say Ab, Bird, and Fly are the predicates,
and suppose there are wffs $\alpha(x), \beta_1(x)$, and $\beta_2(x)$
such that

$KB[Ab/\alpha; Bird/\beta_1; Fly/\beta_2]$ is true
and $\forall x[\alpha(x) \supset Ab(x)]$ is true

then want to conclude by default that

$\forall x[\alpha(x) \equiv Ab(x)]$ is true.

will ensure that Ab is as small as possible

In general:

where Ab_i are the abnormality predicates
and P_i are all the other predicates,

$\text{Circ}(KB)$ is the set of all wffs of the form

$KB[Ab_1/\alpha_1; \dots; Ab_n/\alpha_n; P_1/\beta_1; \dots; P_m/\beta_m]$
 $\wedge \forall x[\alpha_1(x) \supset Ab_1(x)] \wedge \dots \wedge \forall x[\alpha_n(x) \supset Ab_n(x)]$
 $\supset \forall x[\alpha_1(x) \equiv Ab_1(x)] \wedge \dots \wedge \forall x[\alpha_n(x) \equiv Ab_n(x)]$...

KR & R © Brachman & Levesque 2005

Defaults

Circumscription - 2

Theorem: If $KB \cup \text{Circ}(KB) \models \alpha$ then $KB \models_m \alpha$

So this gives us a sound but incomplete method of determining minimal entailments

to get a complete version, would have to use "second order logic," which quantifies over predicates

as in: $\forall \phi [KB[Ab/\phi] \wedge \forall x(\phi(x) \supset Ab(x)) \dots]$

Use: guess at a "minimal" α_i and appropriate other β_i such that $KB \models KB[Ab/\dots] \wedge \forall x[\alpha_i(x) \supset Ab_i(x)]$, then:

- $KB[Ab/\dots] \wedge \forall x[\alpha_i(x) \supset Ab_i(x)] \supset \forall x[\alpha_i(x) \equiv Ab_i(x)]$ is a member of $\text{Circ}(KB)$
- so $KB \cup \text{Circ}(KB) \models \forall x[\alpha_i(x) \equiv Ab_i(x)]$
- since α_i was chosen to be some minimal set of abnormal individuals, it follows from $KB \cup \text{Circ}(KB)$ that these are the only instances of Ab_i
- so any other individual will have the properties of normal individuals

For the bird example, a minimal α is $(x = \text{chilly})$, for which a suitable β_1 is $\text{Bird}(x)$ and β_2 is $(x \neq \text{chilly})$.

$KB \cup \text{Circ}(KB) \models \forall x[(x = \text{chilly}) \equiv \text{Ab}(x)]$

$KB \cup \text{Circ}(KB) \models \neg \text{Ab}(\text{tweety})$

Assignment Project Exam Help

<https://powcoder.com>

Fixed / variable predicates

Imagine KB as before +

$\forall x[\text{Penguin}(x) \supset \text{Bird}(x) \wedge \neg \text{Fly}(x)]$

Get: $KB \models \forall x[\text{Penguin}(x) \supset \text{Ab}(x)]$

so minimizing Ab also minimizes penguins!

Get: $KB \models_m \forall x \neg \text{Penguin}(x)$

McCarthy's definition:

Let **P** and **Q** be sets of predicates

$I_1 \leq I_2$ iff same domain and

1. $\Phi_1(P) \subseteq \Phi_2(P)$, for every $P \in \mathbf{P}$ Ab predicates
2. $\Phi_1(Q) = \Phi_2(Q)$, for every $Q \notin \mathbf{Q}$

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

Get definition of \models_m that is parameterized by what is minimized and what is allowed to vary

need a different definition of $\text{Circ}(KB)$ too

In previous examples, want to minimize Ab while allowing only Fly to vary (so keep Penguin fixed)

Problems:

- need to decide what to allow to vary
- cannot conclude $\neg \text{Penguin}(\text{tweety})$ by default!
only get default $(\neg \text{Penguin}(\text{tweety}) \supset \text{Fly}(\text{tweety}))$

Default logic

Beliefs as deductive theory

explicit beliefs = axioms

implicit beliefs = theorems

least set closed under inference rules

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

Would like to generalize to default rules:

If can prove $\text{Bird}(x)$, but cannot prove $\neg \text{Fly}(x)$,
then infer $\text{Fly}(x)$.

Problem: how to characterize theorems

cannot write down a derivation as before, since we
will 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 extensions

sets of sentences that are "reasonable" beliefs,
given facts and default rules

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Extensions

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

- F is a set of sentences (facts)
- D is a set of default rules: 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

example: $\langle \text{Bird}(\text{tweety}), \text{Fly}(\text{tweety}), \text{Fly}(\text{tweety}) \rangle$

treat $\langle \text{Bird}(x), \text{Fly}(x), \text{Fly}(x) \rangle$ as set of rules

Default rules where $\beta = \gamma$ are called normal

write as $\langle \alpha \Rightarrow \beta \rangle$

will see later a reason for wanting non-normal ones

A set of sentences E is an extension of $\langle F, D \rangle$
iff for every sentence π , E satisfies

$$\pi \in E \text{ iff } F \cup \Delta \models \pi$$

$$\text{where } \Delta = \{ \gamma \mid \langle \alpha, \beta, \gamma \rangle \in D, \alpha \in E, \neg \beta \notin E \}$$

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

$F = \text{Bird}(\text{chilly}), \neg \text{Fly}(\text{chilly}), \text{Bird}(\text{tweety})$
 $D = \langle \text{Bird}(x) \Rightarrow \text{Fly}(x) \rangle$

then there is a unique extension:

$\Delta = \text{Fly}(\text{tweety})$

- Resulting E is an extension since tweety is the only t for this Δ such that $\text{Bird}(t) \in E$ and $\neg \text{Fly}(t) \notin E$.
- No other extension, since the same applies no matter what $\text{Fly}(t)$ assumptions are in Δ .

But in general can have multiple extensions:

$F = \{ \text{Republican}(\text{dick}), \text{Quaker}(\text{dick}) \}$
 $D = \{ \langle \text{Republican}(x) \Rightarrow \neg \text{Pacifist}(x) \rangle, \langle \text{Quaker}(x) \Rightarrow \text{Pacifist}(x) \rangle \}$ conflicting defaults

Have two extensions:

E_1 has $\Delta = \neg \text{Pacifist}(\text{dick})$
 E_2 has $\Delta = \text{Pacifist}(\text{dick})$

Which to believe?

credulous: choose an extension arbitrarily

skeptical: believe what is common to all extensions

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

$\langle \text{Quaker}(x), \text{Pacifist}(x) \wedge \neg \text{Republican}(x), \text{Pacifist}(x) \rangle$

but need to consider all possible interactions in defaults!

KR & R © Brachman & Levesque 2005

Defaults

Assignment Project Exam Help

<https://powcoder.com>

Unsupported conclusions

Definition of extension tries to eliminate facts that do not result from either F or D .

for example, we do not want $\text{Yellow}(\text{tweety})$ and its entailments in the extension

no unsupported conclusions

But the definition has a problem:

Suppose $F = \{ \}$ and $D = \langle p, \text{True}, p \rangle$.

Then $E = \text{entailments of } \{p\}$ is an extension

since $p \in E$ and $\neg \text{True} \notin E$, 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 rule should have no effect

Want unique extension: $E = \text{entailments of } \{ \}$

Reiter's definition:

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

if $\langle \alpha, \beta, \gamma \rangle \in D$, $\alpha \in \Gamma(S)$, and $\neg \beta \notin S$,
 then $\gamma \in \Gamma(S)$.

note: not $\Gamma(S)$

A set E is an extension of $\langle F, D \rangle$ iff $E = \Gamma(E)$.

called a fixed point of the Γ operator

KR & R © Brachman & Levesque 2005

Defaults

Autoepistemic logic

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

$$\langle \alpha, \beta, \gamma \rangle \not\models \langle \alpha, \beta, (\gamma \vee \delta) \rangle$$

Solution: express defaults as sentences in extended language that talks about belief

for any sentence α , have another sentence $\mathbf{B}\alpha$

$\mathbf{B}\alpha$ says "I believe α ": autoepistemic logic

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

any bird not believed to be flightless flies

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

modal logic of belief provide semantics

for here: treat $\mathbf{B}\alpha$ as if it were an new atomic wff

$$\text{still get: } \forall x[\text{Bird}(x) \wedge \neg \mathbf{B}\neg \text{Fly}(x) \supset \text{Fly}(x) \vee \text{Run}(x)]$$

Main property for set of implicit beliefs, E :

1. If $E \models \alpha$ then $\alpha \in E$. (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 stable

KR & R

© Brachman & Levesque 2005

Defaults

<https://powcoder.com>

Add WeChat powcoder

Stable expansions

Given KB, possibly containing \mathbf{B} operators, what is an appropriate stable set of beliefs?

want a stable set that is minimal

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

$$\pi \in E \text{ iff } \text{KB} \cup \Delta \models \pi,$$

$$\text{where } \Delta = \{\mathbf{B}\alpha \mid \alpha \in E\} \cup \{\neg \mathbf{B}\alpha \mid \alpha \notin E\}$$

fixed point of another operator

analogous to the extensions of default logic

Example:

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

get a unique stable expansion containing $\text{Fly}(\text{tweety})$

As in default logic, stable expansions are not uniquely determined

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

2 stable expansions: one with p , one with q

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

no stable expansions – so what to believe?

KR & R

© Brachman & Levesque 2005

Defaults

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, then we can enumerate all stable expansions using the following:

1. Suppose $\mathbf{B}\alpha_1, \mathbf{B}\alpha_2, \dots, \mathbf{B}\alpha_n$ are all the **B** wffs in KB.
2. Replace some of these by True and the rest by $\neg\text{True}$ 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 $\text{KB}^\circ \models \alpha_i$
 - if $\mathbf{B}\alpha_i$ was replaced by $\neg\text{True}$, then $\text{KB}^\circ \not\models \alpha_i$
4. If yes, then KB° determines a stable expansion.
entailments of KB° are the objective part

Example:

For $\text{KB} = \{\text{Bird}(\text{chilly}), \neg\text{Fly}(\text{chilly}), \text{Bird}(\text{tweety}), [\text{Bird}(\text{tweety}) \wedge \neg\mathbf{B}\neg\text{Fly}(\text{tweety}) \supset \text{Fly}(\text{tweety})], [\text{Bird}(\text{chilly}) \wedge \neg\mathbf{B}\neg\text{Fly}(\text{chilly}) \supset \text{Fly}(\text{chilly})]\}$

Two **B** wffs: $\mathbf{B}\neg\text{Fly}(\text{tweety})$ and $\mathbf{B}\neg\text{Fly}(\text{chilly})$,
so four replacements to try

Only one works: $\mathbf{B}\neg\text{Fly}(\text{tweety}) \mapsto \neg\text{True}$
 $\mathbf{B}\neg\text{Fly}(\text{chilly}) \mapsto \text{True}$

Resulting KB° has $(\text{Bird}(\text{tweety}) \supset \text{Fly}(\text{tweety}))$

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

More ungroundedness

Definition of stable expansion may not be strong enough

$\text{KB} = \{(\mathbf{B}p \supset p)\}$ has 2 stable expansions:

- one without p and with $\neg\mathbf{B}p$
corresponds to $\text{KB}^\circ = \{\}$
- one with p and $\mathbf{B}p$.
corresponds to $\text{KB}^\circ = \{p\}$

But why should p be believed?

only justification for having p is having $\mathbf{B}p$!
similar to problem with default logic extension

Konolige's definition:

A grounded 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 exact equivalence with Reiter's definition of extension in default logic