Closed World Reasoning

Yulia Levin

Today :

- Introduction: Default reasoning, Monotonic vs. Non-monotonic reasoning, Closed world reasoning vs. open world
- Formal approaches to closed world reasoning
- Logic Programming and NAF

Introduction- Monotonic vs. Non monotoning reasoning

Monotonic reasoning:

New facts can only produce additional beliefs.

Formally:

If $T \models A$ and $T \subseteq S$ then $S \models A$.

Introduction- Monotonic vs. Non monotoning reasoning

<u>Monotonic reasoning</u>

<u>Example:</u>

Suppose we have a database containing information about animals.

We would like to find out whether a certain individual “Bobik” is a carnivore.

Assuming the data we have contains the sentence Dog(Bobik) .

There are only two possibilities for getting the conclusion Carnivore (Bobik):

1. **The database contains other facts that mention the constant Bobik explicitly.**
2. **The database entails a universal of the form**

$$\forall x. Dog(x) -> Carnivore(x)$$

Default Reasoning

Just as before we would like to find out whether Bobik is a Carnivore given: Dog(Bobik).

Assuming we know that dogs are, generally speaking, carnivores. Then in order to find the answer we shall follow the rule:

Given that a P is generally a Q, and given that P(a) is true, it is reasonable to conclude that Q(a) is true unless there is a good reason not to.

Default Reasoning

The definition is somewhat vague since the "good reason" part is not defined. However we often use default reasoning:

If we are asked for the color of a polar bear

The answer by default would be – white.

If we happen to know that the current bear has been rolling in mud, we might change our mind.

Default Reasoning- when?

General statements:

- Normal :Under typical circumstances, Ps are Qs.
  (People work close to where they live)
- Prototypical: The prototypical P is a Q.
  (Tomatoes are red)
- Statistical: Most Ps are Qs
  (The people in the waiting room are growing impatient)

Default Reasoning- when?

Lack of information to the countrary:

- Familiarity: If a P was not a Q, you would know it.

  (No nation has a political leader more than 7 feet tall)
- Group confidence: All known Ps are known (or assumed) to be Qs.

  (Natural languages are easy for children to learn)

Default Reasoning- when?

Conventional uses:

Conversational: A P is a Q,Unless I tell you otherwise.

(Being told “ The closest gas station is two blocks east,” the assumed default is that the gas station is open)

Representational: A P is a Q, unless otherwise indicated.

(The speed limit in the city)

As we can see, default reasoning is non-monotonic.

Introduction- Monotonic vs. Non monotoning reasoning

<u>Nonmonotonic reasoning:</u>

New facts will sometimes invalidate previous beliefs.

For example: **Recall tweety**

If we are only told that tweety is a bird we may conclude that tweety flies. However if we are now told that tweety is an emu / ostrich/ penguin we may no longer believe that she flies.

Closed world reasoning

The simplest formalization of default reasoning is called the closed world reasoning.

Observation:

The number of negative facts about a given domain is typically much greater than the number of the positive ones.

In many natural applications, the number of negative facts is so large that their explicit representation becomes practically impossible.

Closed world reasoning

Example: The data base of a lending – library. Suppose there are 1000 readers and 10000 books, and each reader is allowed to borrow up to 5 books. In order to keep track of all readers and all the books they do **not** currently borrow(the negative facts)

we would need to store a huge amount of data.

Closed world reasoning
The closed world assumption

The natural solution to this problem is to assume that all positive information has been specified, and conclude that any positive fact that has not been specified or cannot be inferred from this information is false.

This is precisely the CWA rule.

Closed world assumption vs. open world assumption

CWA (Closed world Assumption):

Complete knowledge is assumed -

Unless an atomic sentence is known to be true, it can be assumed to be false.

OWA (Open World Assumption):

We assume that our knowledge of the world might be incomplete and therefore any information not explicitly specified(or such that cannot be derived from the known data) is considered unknown .

Closed world assumption vs. open world assumption

Example:

Given : Yossi is a citizen of Israel

Question: Is Yossi a citizen of USA?

Answer: CWA – No.

OWA – Unknown

(Yossi could have dual citizenship)

Logic framework - reminder

- Clause – A disjunction of literals.
- Horn Clause – A clause with at most one positive literal.
- Definite Horn Clause – A Horn clause with exactly one positive literal.

Example: This is a Definite Horn Clause:

$(\neg p \vee \neg q \vee .... \vee \neg t \vee u)$ which can also be written as: $(p \wedge q \wedge ... \wedge t) \rightarrow u$

Logic framework - reminder

- Herbrand Universe of a theory T –

  The set of all ground terms formed using the function and individual constants occurring in the theory T (we assume that each theory has at least one individual constant)

Logic framework - reminder

- The Herbrand base of T – HB(T) – the set of all sentences of the form $P(\alpha_1,...,\alpha_n)$ where P is a predicate constant occurring in T and $\alpha_1,...,\alpha_n$ are elements of the Herbrand Universe of T.

Logic framework - reminder

- We say that a theory T is consistent iff there is no sentence α such that both ¬α and α are known.
- We say that a theory T exhibits complete knowledge iff for every sentence α (within its vocabulary) ,either α or ¬α are known.

CWA

- The first formalization of the CWA rule was introduced by Reiter (1978)
- Reiter proposed the following syntactic formalization:

$$CWA(T) = T \cup \{\neg A : T \nvdash A \quad and \quad A \in HB(T)\}$$

CWA

<u>Example:</u>

Let T be:

$$\forall x.P(x) \to Q(x)$$

$$P(a) \wedge R(b)$$

$$HB(T) = \{P(a), Q(a), R(a), P(b), Q(b), R(b)\}$$

$$CWA(T) = T \cup \{\neg R(a), \neg P(b), \neg Q(b)\}$$

CWA

- Thus applying CWA on our database we achieve complete knowledge as defined above.
- According to the formalization of CWA(T) as presented above we get that if CWA(T) is consistent and A is a ground positive sentence, then

1. $CWA(T) \vdash A \quad \text{iff} \quad T \vdash A$
2. $CWA(T) \vdash \neg A \quad \text{iff} \quad T \nvdash A$

CWA - consistency

- The CWA as presented doesn't necessarily preserve consistency.

  Motivating Example:

  Let $T = \{ Student(Yossi) \vee Student(Moshe) \}$

  $$CWA(T) = T \cup \{ \neg Student(Yossi), \neg Student(Moshe) \}$$

  is inconsistent!

CWA - consistency

- Theorem: Let T be a consistent set of formulas. The theory CWA(T) is consistent iff any clause $P_1 \vee \ldots \vee P_n$ that is formed of positive ground literals P_i and that can be deduced from T, contains at least one positive ground literal P_i that can be deduced from T.
- If T is a consistent Horn theory, then CWA(T) is consistent.

Logic framework - reminder

Definition: A Herbrand frame of a theory T is

Any frame M such that:

- |M| is the Herbarnd universe for T.
- For each individual constant c occurring in T, M|c| = c.
- For each n-ary function constant f occuring in T, M|f| is the function which assigns the ground term $f(a_1,\ldots,a_n)$ to a tuple $a_1,\ldots,a_n$ of ground terms.

Logic framework - reminder

- Since Herbrand frames only differ in how they interpret predicate constants, each such a frame can be identified with a subset of HB(T).

  For Example: The set $\{P(a), Q(a, f(a))\}$

  corresponds to the Herbrand frame M with

  M|P| = {a}, M|Q| = {(a, f(a))}

Logic framework

- A Herbarand Model of a theory T is a Herbrand farme in which all sentences of T are True.
- A minimal Herbrand Model of a theory T is such that any subset is not a model.
- The least Herbrand Model of a theory T is such that any other Herbrand Model is a superset.
  - May not exist: $P(a) \vee P(b)$

CWA - semantics

- Theorem: For each theory T , CWA(T) is consistent iff T has the least Herbrand model.
- Each consistent Horn theory has the least Herbrand Model
- If T is a consistent Horn theory, then CWA(T) is consistent.
- If CWA(T) is consistent, then for any ground sentence A, CWA(T) $\vdash$ A iff A is true in the least Herbrand model of T.

Additional Conventions Dealing with quantifiers

Motivating Example: Airline flight guide

- A Language L – contains a single predicate Connect and constants $c_1, \ldots, c_n$.
- A theory T in the above language containing only atomic sentences of the form Connect(c_i, c_j).
  Connect (London, Tel – Aviv)
  Connect (Paris, Tel – Aviv)

Additional Conventions Dealing with quantifiers

Suppose there is a city "smallTown" that has no airport and doesn't appear in our guide. Therefore applying CWA on T we get that the sentence

¬Connect(c_i ,SmallTown) is entailed.

However niether the sentence

$$\neg\exists x.\mathbf{Connect(x,SmallTown)}$$

Nor its negation are entailed.

Additional Conventions Dealing with quantifiers

The reason is that the domain might include a city not named by any c_i.

The solution would be adding the *domain closure assumption*: there are no other individuals than those in the database

Formally: $\forall x((x = c_1) \vee \ldots \vee (x = c_n))$

Additional Conventions Dealing with equality

Including the *unique name assumption :* different names represent different objects.

We also get the ability to deal with equality.

For languages with no function constants this assumption can be presented by:

$$\left\{ \begin{array}{l} \neg(c_i = c_j) : c_1 \ldots c_n \textbf{ are the individual constants of the language} \\ \qquad \textit{and } 1 \le i < j \le n \end{array} \right\}$$

Additional Conventions
Dealing with equality

Since we have introduced the equality predicate we must also introduce the axioms that represent its qualities: (for languages without function constants)

- Reflexivity - $\forall x(x=x)$
- Symmetry - $\forall x \forall y\ ((x=y) \rightarrow (y=x))$
- Transitivity - $\forall x \forall y \forall z\, ((x=y) \wedge (y=z) \rightarrow (x=z))$
- Principle of substitution of equal terms- $\forall x_1 \ldots \forall x_n \forall y_1 \ldots \forall y_n (P(x_1,...,x_n) \wedge (x_1 = y_1) \wedge ... \wedge (x_n = y_n) \rightarrow$ $P(y_1,...,y_n))$

Generalized Closed World Assumption (GCWA)

- GCWA – A consistency preserving extension of CWA (Introduced by Minker 1982).
- Motivating Example: Let T be

$$\mathbf{Course(Logica)} \wedge \mathbf{Course(Modelim)}$$

$$\mathbf{Student(Yossi)} \vee \mathbf{Teacher(Yossi)}$$

Applying CWA on T would lead to inconsistency

Generalized Closed World Assumption (GCWA)

However it’s still reasonable (and consistency preserving) to assume that “Logica” and “Modelim” are neither students nor teachers, and that Yossi is not a course…

Generalized Closed World Assumption (GCWA)

Definition: Let T be a theory. A ground atom $A \in HB(T)$ is *free for negation in T* iff there is no clause $C = C_1 \vee \ldots \vee C_n$, where each $C_i \in HB(T)$ such that:

1. $T \vdash A \vee C$
2. $T \nvdash C$

We denote by NFREE(T) the set of all atoms Which are free for negation in T.

Generalized Closed World Assumption (GCWA)

So an atom A can be assumed to be false only if it is the case that whenever a disjunction of atoms including that atom is entailed by T the smaller disjunction without the atom is also entailed.

Thus the generalized closure of a theory T is:

$$\mathbf{GCWA(T) = T \cup \{\neg A : A \in NFREE(T)\}}$$

Generalized Closed World Assumption (GCWA)

Example:

$$T = \left\{ \begin{array}{l} \text{Course(Logica)} \wedge \text{Course(Modelim)} \\ \text{Student(Yossi)} \vee \text{Teacher(Yossi)} \end{array} \right\}$$

$$\text{GCWA(T)} = \left\{ \begin{array}{l} \neg\text{Course(Yossi)}, \neg\text{Student(Logica)}, \neg\text{Student(Modelim)} \\ \neg\text{Teacher(Logica)}, \neg\text{Teacher(Modelim)} \end{array} \right\}$$

Generalized Closed World Assumption (GCWA)

- If T is a Horn theory, then GCWA(T) = CWA(T).
- For each theory T and each positive ground sentence A, GCWA(T) $\vdash$ A iff T $\vdash$ A.
- If T is consistent, then GCWA(T) is also consistent.
- For each theory T and each $\mathbf{A \in HB(T)}$,

  $\mathbf{A \in NFREE(T)}$ iff ¬A is true in each minimal Herbrand model of T

Generalized Closed World Assumption (GCWA)

- We shall write $T \models_m A$ to indicate that a formula A is true in all minimal Herbrand models of T.
- Theorem: For each theory T and each ground literal A, $GCWA(T) \vdash A$ iff $T \models_m A$.

The last theorem doesn't hold for arbitrary ground sentences!

Generalized Closed World Assumption (GCWA)

Example: $T = \{\mathbf{Bird(Tweety)} \vee \mathbf{Bird(Yossi)}\}$

$M_1 = \{\mathbf{Bird(Tweety)}\}$ – minimal

$M_2 = \{\mathbf{Bird(Yossi)}\}$ – minimal

$M_3 = \{\mathbf{Bird(Tweety)}, \mathbf{Bird(Yossi)}\}$

Consider $A = \neg(\mathbf{Bird(Tweety)} \wedge \mathbf{Bird(Yossi)})$

A is True in both the minimal Herbarnd models of T. However, A cannot be derived using GCWA.

Careful CWA (CCWA)

CCWA is an extension of GCWA developed by Gelfond & Przymusinska(1986).

The new feature: It allows us to restrict the effects of closing the world by specifying the predicates which may be affected by the CWA rule. Other predicates also specified by the user, are permitted to vary in the process of closure.

Careful CWA (CCWA)

The set of all predicate constants occuring in a given theory will be devided into three disjoint groups: $\vec{\mathbf{P}}$, $\vec{\mathbf{Q}}$, $\vec{\mathbf{R}}$

- For a tuple $\vec{\mathbf{S}}$ of predicate constants in T denote $\vec{\mathrm{S}}^{+}$ the set of all positive ground literals constructible using predicate constants from $\vec{\mathrm{S}}$

Careful CWA (CCWA)

- Let T be a theory. A ground atom $A \in HB(T)$ is *free for negation in T wrt to* $\vec{P}, \vec{Q}, \vec{R}$ iff $A \in \vec{P}^+$ and there is no clause $C = C_1 \vee ... \vee C_n$, where each $C_i \in \vec{P}^+ \cup \vec{R}^+ \cup \vec{R}^-$ such that:

1. $T \vdash A \vee C$
2. $T \nvdash C$

We denote by $NFREE(T;\vec{P};\vec{Q};\vec{R})$ the set of all atoms from $\vec{P}^+$ which are free for negation in T.

Careful CWA (CCWA)-Example

$$T = \{ Bird(Tweety) \wedge (\forall x. Bird(x) \wedge \neg Ab(x) \rightarrow Flies(x)) \}$$

$$\vec{P} = (Ab), \vec{Q} = (Flies), \vec{R} = (Bird)$$

$$\vec{P}^{+} = \{ Ab(Tweety) \}$$

$$\vec{P}^{+} \cup \vec{R}^{+} \cup \vec{R}^{-} = \{ Ab(Tweety), Bird(Tweety), \neg Bird(Tweety) \}$$

$$CCWA(T) = T \cup \{ \neg Ab(Tweety) \}$$

$$CCWA(T) \models Flies(Tweety)$$

Extended CWA(ECWA)

ECWA is the most powerful of all CWA logics, Developed by Gelfond (1989).

The new feature:

The new formalism augments the theory under consideration with ground sentences, rather than ground atoms.

Logic Programming - Prolog

Logic Programming can be described as the use of mathematical logic for computer programming.

Prolog is an example for a logic programming system.

A prolog program is made up of a sequence of Definite Horn clauses.

Logic Programming – Prolog – negative expressions

Prolog allows the use of negative expressions using a special operator “not”.

This is a predefined predicate which is based upon the ***negation as failure principle***.

Negation as Failure (NAF)

NAF is a non-monotonic inference rule in logic programming used to derive $\neg \mathbf{p}$ from the failure to derive p.

Basically: Given a ground positive literal p

not(p) evaluates to True if we cannot find a

finite proof for p from the information

contained in the system; otherwise it would

take the value False.

Negation as Failure (NAF)

- The inclusion of negation as failure means that logic programming is a kind of non-monotonic logic.

Prolog “not” predicate example

fly(x) : - bird(x), not(abnormal(x)).

abnormal(x) : - ostrich(x).

bird(Tweety).

? – abnormal(tweety).

---> no

? – fly(tweety).

---> yes

Prolog “not” predicate example (cont)

- Adding information might change the outcome:

  fly(x) : - bird(x), not(abnormal(x)).

  abnormal(x) : - ostrich(x).

  bird(Tweety).

  ostrich(tweety).

  ? – abnormal(tweety).

  ---> yes

  ? – fly(tweety).

  ---> no

NAF - CWA

NAF and CWA have a common idea:
If a positive ground literal cannot be "proved", then its negative form can be interpreted as True.
The difference: CWA is based on the concept of pure logical proof, whereas negation as failure relies upon the concept of proof that underlies the Prolog algorithm.

Theory completion

- In 1978 Keith Clark tried to resolve the logical status of negation as failure by showing that under certain conditions NAF is an implementation of classical negation with respect to the *completion of the program.*

Theory Completion

Theory completion is based upon the principle illustrated in the following example:

“If I arrive after nine o’clock , then I am late”

vs.

“I am late if and only if I arrive after nine o’clock”

Theory Completion

- A rule is any formula of the form: $\mathbf{B_1} \wedge \ldots \wedge \mathbf{B_n} \rightarrow \mathbf{A}$

  if A is P(a) we say that the rule is about P.
- Program – A theory T that contains only a finite set of rules.
- If $\mathbf{B_1, \ldots, B_n}$ are positive, then the rule is definite.
- A program is definite iff each of its rules is definite.
- *Completion of a program* T consists in completing T with respect to every predicate P for which there exists a rule about.

Theory Completion

$$1. \text{False} \rightarrow Q(x_1, \ldots, x_k)$$

$$2. B_1 \wedge \ldots \wedge B_n \rightarrow P(\alpha_1 \ldots \alpha_k)$$

$$B_1 \wedge \ldots \wedge B_n \wedge (x_1 = \alpha_1) \wedge \ldots \wedge (x_k = \alpha_k) \rightarrow P(x_1 \ldots x_k)$$

$$3. C_1 \rightarrow P(x_1^1 \ldots x_k^1) \qquad C_1 \rightarrow P(x_1 \ldots x_k)$$

$$C_p \rightarrow P(x_1^p \ldots x_k^p) \qquad C_p \rightarrow P(x_1 \ldots x_k)$$

$$4. \exists y_1 \ldots y_n C_j \rightarrow P(x_1 \ldots x_k)$$

$$5. \forall x_1 \ldots x_k [(D_1 \vee \ldots \vee D_p) \rightarrow P(x_1, \ldots, x_k)]$$

$$6. \forall x_1 \ldots x_k [(D_1 \vee \ldots \vee D_p) \leftrightarrow P(x_1, \ldots, x_k)]$$

Theory Completion- Example

$$T=\left\{\begin{array}{l} \mathrm{Adult(Yossi)} \\ \forall x.\ \mathrm{Student}(x) \ \rightarrow \mathrm{Adult}(X) \\ \forall x.\mathrm{Adult}(x) \wedge \neg \mathrm{Student}(x) \rightarrow \mathrm{Employed}(x) \end{array}\right\}$$

$$\mathrm{COMP}(T)=\left\{\begin{array}{l} \forall x.\mathrm{Adult}(x) \leftrightarrow \mathrm{Student}(x) \ \vee (x=\mathrm{Yossi}) \\ \forall x.\ \mathrm{Employed}(x) \leftrightarrow \mathrm{Adult}(x) \ \wedge \neg \mathrm{Student}(x) \\ \forall x. \neg \mathrm{Student}(x) \end{array}\right\}$$

- COMP(T) is acctually what stated above together with the equality axioms and the unic name assumption

Theory Completion- Example

$$T' = \left\{ \begin{array}{l} \mathrm{Adult(Yossi)} \\ \forall x.\ \mathrm{Student}(x) \rightarrow \mathrm{Adult}(X) \\ \forall x.\ \mathrm{Adult}(x) \wedge \neg \mathrm{Employed}(x) \rightarrow \mathrm{Student}(x) \end{array} \right\}$$

$$\mathrm{COMP}(T') = \left\{ \begin{array}{l} \forall x.\ \mathrm{Adult}(x) \leftrightarrow \mathrm{Student}(x) \vee (x = \mathrm{Yossi}) \\ \forall x.\ \mathrm{Student}(x) \leftrightarrow \mathrm{Adult}(x) \wedge \neg \mathrm{Employed}(x) \\ \forall x.\ \neg \mathrm{Employed}(x) \end{array} \right\}$$

$$\boxed{\begin{array}{l} \mathrm{COMP}(T) \vdash \mathrm{Employed(Yossi)} \\ \mathrm{COMP}(T') \vdash \neg \mathrm{Employed(Yossi)} \end{array}}$$

Theory Completion

- We have seen that logically equivalent programs may have different completions.
- COMP(T) doesn’t necessarily preserve consistency!
- COMP(T) for definite programs preserves consistency.
- For Definite Programs:
  - For any ground sentence A,

$$\mathbf{COMP(T)} \vdash \mathbf{A} \Rightarrow \mathbf{CWA(T)} \vdash \mathbf{A}$$

  - For any ground positive sentence A

$$\mathbf{COMP(T)} \vdash \mathbf{A} \ \textit{iff} \ \mathbf{T} \vdash \mathbf{A}$$