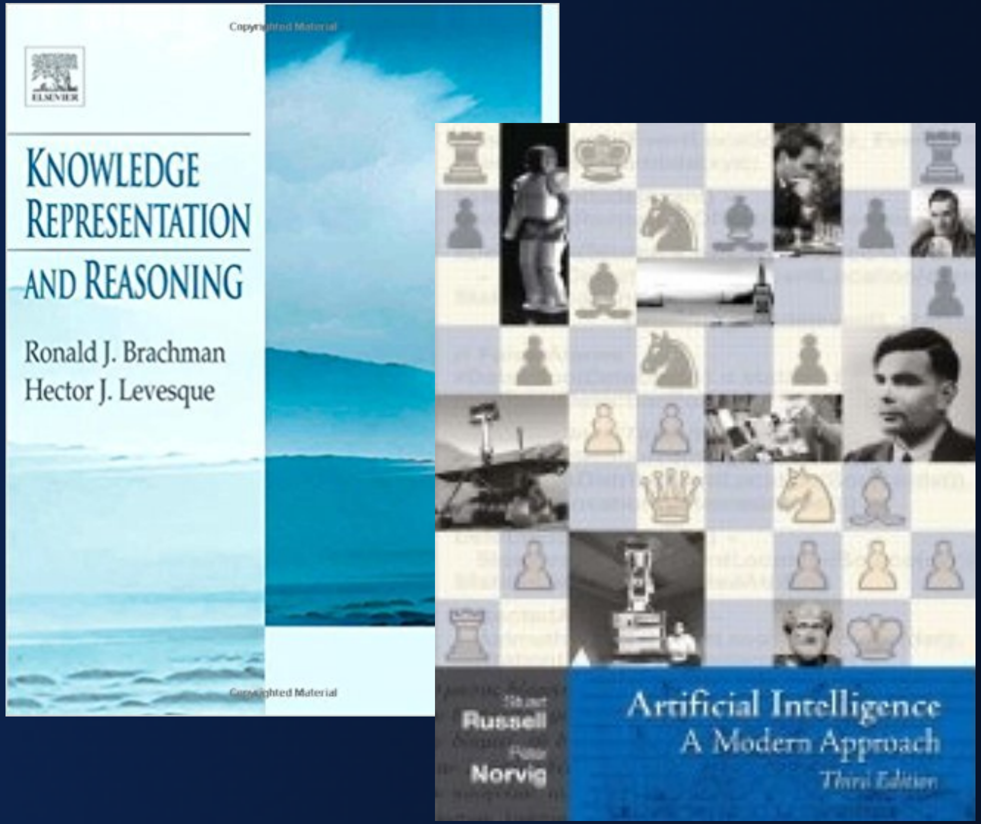


AI Fundamentals: Knowledge Representation and Reasoning

Maria Simi

Nonmonotonic reasoning

LESSON 3: CLOSED WORLD ASSUMPTION – CIRCUMSCRIPTION – DEFAULT LOGICS

Monotonicity of classical logic

Classical entailment is **monotonic.**

If $KB \vDash a$, then $KB \cup \{b\} \vDash a$ $\quad [KB \land b \vDash a]$

Failures of monotonicity are widespread in *commonsense reasoning*. It seems that humans often "jump to conclusions", when they think it is safe to do so (lacking information to the contrary).

These conclusions are only "reasonable", given what you know, not **classically entailed**.

Most of the inference we do is *defeasible*: additional information, may lead to to retract those tentative conclusions. The set of beliefs does not grow monotonically as new evidence arrives; the monotonicity property is violated.

.

Instances of nonmonotonic reasoning

Some common instances of nonmonotonic reasoning:

1. Default reasoning: reasonable assumptions unless evidence of the contrary
   - Car parked on the street; you assume it has four wheels even if you can see only two.
   - Birds fly, swan are white, bananas are yellow, tomatoes are red (prototypes).
2. Persistence: things stay the same, according to a principle of inertia, unless we know they change
3. Economy of representation: only true facts are stored, false facts are only assumed
4. Reasoning about knowledge: if you have $\neg Know(p)$ and you learn p ...
5. Abductive reasoning: most likely explanations to known facts.

Strictness of FOL universals

Universal rules, e.g. $\forall x\,(P(x) \Rightarrow Q(x))$

- express properties that apply to all instances
- all or nothing!

But most of what we learn about the world is in terms of **generics** rather than **universals**

Encyclopedia entries for ferry wheels, violins, turtles, wildflowers

E.g. “*Violins have four strings*” vs “*All violins have four strings*”

Properties are not strict for all instances, because

- genetic / manufacturing varieties
- borderline cases (early ferry wheels – toy violins)
- cases in exceptional circumstances etc.

Universal with exceptions

Listing exceptions is not a viable solution:

- *"All violins that are not* E_1 *or* E_2 *or … have four strings"*.
  Exceptions usually are difficult to enumerate: *qualification problem.*
  Similarly, for general properties of individuals.
- Goal: be able to say a P is a Q in general, normally, but not necessarily. It is reasonable to conclude $Q(a)$, given $P(a)$, *unless there is a good reason not to.*
- This is what we call a **default** and **default reasoning** the tentative conclusion**.**
- Note: qualitative version (no numbers involved)

Approaches

There are two ways to approach the problem.

1. **Model theoretic formalizations** (CWA, Circumscription):
   - consist in a restriction to the possible interpretations, redefining the notion of entailment;
   - we can still have systems sound and complete *wrt* the new semantics.
2. **Proof theoretic formalizations** (Default logic, Autoepistemic logic)
   - A proof system with nonmonotonic inference rules

What's next

We will consider four approaches to default reasoning:

1. Closed-world reasoning, i.e. under the Closed World Assumption (CWA)
2. Circumscription
3. Default logic
4. Autoepistemic logic

Then discuss systems supporting **belief revision:**

1. TMS, ATMS

Closed World Assumption (CWA)

- Reiter's observation:

  "There are usually many more negative facts than positive facts!"

  Example: airline flight database provides:

  $DirectConnect(cleveland, toronto)$ $DirectConnect(toronto, northBay)$,

  $DirectConnect(toronto, winnipeg)$ …

  but not: $\neg DirectConnect(cleveland, northBay)$ …

  The classical logical answer to $DirectConnect(cleveland, northBay)$: "I don't know"
- Under **Closed World Assumption (CWA)** only positive facts are stored, any other fact is assumed false.

  The answer to $DirectConnect(cleveland, northBay)$ under CWA : "No"
- CWA assumption is used in databases and in logic programming with negation as failure.

Complete and incomplete knowledge

- CWA can be seen as an assumption about **complete knowledge**

KB with consistent knowlegde:

For no α , $KB \vDash \alpha$ and $KB \vDash \neg\alpha$

KB with complete knowledge:

For every α, $KB \vDash \alpha$ or $KB \vDash \neg\alpha$

- Normally, a KB has incomplete knowledge:

Let $KB = \{p \vee q\}$

Then $KB \vDash (p \vee q)$

But for p : $KB \nvDash p$ and $KB \nvDash \neg p$

Similarly for q: $KB \nvDash q$ and $KB \nvDash \neg q$

Similarly for any ground atom not mentioned in KB .

Semantics and properties of CWA

- CWA corresponds to a new version of entailment:

  Def CWA: $KB \models_c a \text{ iff } KB^+ \models a$

  where $KB^+ = KB \cup \{\neg p \mid p$ ground atom and $KB \not\models p\}$

- By an **inductive argument**, it can be proved that:

  Theorem: For every α (without quantifiers), $KB \models_c \alpha$ or $KB \models_c \neg\alpha$ (1)

Inductive argument:

– immediately true for ground atomic sentences

– $KB \models \neg\neg a \text{ iff } KB \models a$

– $KB \models (a \wedge b) \text{ iff } KB \models a$ and $KB \models b$

– Say $KB \not\models_c (a \vee b)$. Then $KB \not\models_c a$ and $KB \not\models_c b$.

So by induction, $KB \models_c \neg a$ and $KB \models_c \neg b$. Thus, $KB \models_c \neg(a \vee b)$.

Query evaluation

With CWA we can reduce queries (without quantifiers) to atomic queries, by repeated applications of the following properties:

1. $KB \models_c (a \wedge b)$ iff $KB \models_c a$ and $KB \models_c b$
2. $KB \models_c \neg\neg a$ iff $KB \models_c a$
3. $KB \models_c \neg(a \vee b)$ iff $KB \models_c \neg a$ and $KB \models_c \neg b$
4. $KB \models_c (a \vee b)$ iff $KB \models_c a$ or $KB \models_c b$ *for KB completeness*
5. $KB \models_c \neg(a \wedge b)$ iff $KB \models_c \neg a$ or $KB \models_c \neg b$ *for KB completeness*

If KB^+ is consistent, any query reduces to a set of atomic queries:

$KB \models_c p$, where p is an atom

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

Much more efficient than ordinary logic reasoning (e.g. no reasoning by cases).

Consistency of KB^+

Is KB^+ always consistent when KB is consistent? NO.

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

e.g. $KB = \{p \vee q\}$ $\quad KB^+ = KB \cup \{\neg p, \neg q\}$

KB^+ is inconsistent and so for every α, $KB^+ \models \alpha$!

Solution: restrict CWA to atoms that are “uncontroversial”

Def Generalized CWA (GCWA):

$KB^* = KB \cup \{\neg p \mid \text{if } KB \models (p \vee q_1 \vee \ldots \vee q_n) \text{ then } KB \models q_i\}$

For every positive ground literal clause $(p \vee q_1 \vee \ldots \vee q_n)$ entailed by KB, at least one ground literal q_i is also entailed.

Theorem: KB is consistent *iff* the augmentation under GWCA, KB^*, is consistent. (2)

Moreover everything entailed under GCWA is entailed under CWA

Application of GCWA

The application of the theorem of consistency of GCWA (Theorem 2) depends on the terms that we allow as part of the language.

Example: $\boldsymbol{KB} = \{P(x) \vee Q(x), P(A), Q(B)\}$ and the only constants are A *and* B, then KB^* is consistent. If we admit C, then it is not.

The **Domain Closure Assumption (DCA)** may be used to restrict the constants to those explicitly mentioned in the KB.

$\forall x\,.\,[x{=}c_1 \vee \ldots \vee x{=}c_n]$ where c_i are all the **finite** constants appearing in KB

Under this restriction quantifiers can be replaced by finite conjunctions and disjunctions.

The **Unique Names assumption** (**UNA**) applied to term equality is a consequence of the CWA : $(c_i \neq c_j)$, for $i \neq j$

Horn KB consistency under GCWA

Since it may be difficult to test the conditions of **Theorem 2** the following corollary, which restricts the application, is also of practical importance:

Corollary: If the clause form of KB is Horn and consistent, them KB* is consistent.

Remember

Any FOL formula can be transformed into a set of clauses, preserving satisfiability.

Clause: a disjunctions of atomic formulas (positive and negative literals)

$\{l_1, l_2, \ldots l_k\}$

Horn clause: at most one of the literals is positive. Horn clauses take one of two forms:

$\{l_1, \neg l_2, \ldots \neg l_k\}$ or $\{\neg l_2, \ldots \neg l_k\}$

A KB of Horn clauses corresponds to a set of rules.

Predicate completion

The CWA is too strong for many applications, We do not want to assume that **any** ground atom not provable from the KB is false.

Predicate completion has been proposed to address this issue. Certain predicates are considered complete, others are not.

CWA wrt to a predicate P [set of predicates **P**]: the set of *assumed beliefs* is only for ground atoms in P [predicates in **P**].

The theory accounting for **if and when** this leads to consistent augmentation is quite complex, but this is an option to consider. See for example [Genesereth&Nillson, ch 6.2].

Circumscription: minimizing abnormality

- **Circumscription** can be seen as a more powerful and precise version of the CWA, working also for FOL. The idea is to specify special **abnormality predicates** for dealing with exceptions to defaults.

  For example, suppose we want to assert the default rule "*birds fly*":

  $Bird(tweety), Bird(chilly), \neg Flies(chilly), chilly \neq tweety$

  $Bird(x) \wedge \neg Abnormal_1(x) \Rightarrow Flies(x)$ *all normal birds fly*

  We want to derive $Flies(tweety)$, but Tweety could be $Abnormal_1$ in some model

- The solution is to make abnormality predicates "*as false as possible*"

  Circumscription: Given the unary predicate Ab, consider only interpretations where $\mathcal{I}[Ab]$ is **as small as possible**, relative to KB.

  Note: Circumscription is a semantic notion based on **minimal models** (a kind of **model preference logics**) due to MacCarthy 1980.

Minimal entailment

Let **P** be a set of unary abnormality predicates.

Let $\mathcal{I}_2$ and $\mathcal{I}_2$ two interpretations that agree on the values of constants and functions.

Ordering on interpretations:

$\mathcal{I}_1 < \mathcal{I}_2$ *iff* same domain and for every $P \in$ **P** $\mathcal{I}_1[P] \subset \mathcal{I}_2[P]$ holds

Minimal entailment:

$KB \vDash_{\leq} \alpha$ *iff* for every interpretation $\mathcal{I}$, if $\mathcal{I}[KB] = true$ and such that there is no other interpretations $\mathcal{I}' < \mathcal{I}$ such that $\mathcal{I}'[\mathrm{KB}] = true$, then α is true in $\mathcal{I}$.

In simpler words, α must be true in **all interpretations** satisfying KB that minimize abnormalities, those that *most normal*.

Example

Going back to the example:

$Bird(tweety)$, $Bird(chilly)$, $\neg Flies(chilly)$, $chilly \neq tweety$

$Bird(x) \land \neg Ab(x) \Rightarrow Flies(x)$ *all normal birds fly*

$KB \not\models Flies(tweety)$. However, $KB \models_{\leq} Flies(tweety)$

The reason is this:

If $\mathcal{I}[KB] = true$ but $\mathcal{I}[Flies(tweety)] = false$, then $\mathcal{I}[Ab(tweety)] = true$.

So let $\mathcal{I}'$ be exactly $\mathcal{I}$ except that we remove the denotation of $tweety$ from the interpretation of Ab. Then $\mathcal{I}' < \mathcal{I}$ (assuming **P** = $\{Ab\}$), and still $\mathcal{I}'[KB] = true$.

Thus, in the minimal models of the KB, Tweety is a normal bird:

$KB \models_{\leq} \neg Ab(tweety)$, and $KB \models_{\leq} Flies(tweety)$.

We cannot do the same for $chilly$ and in fact $\neg Flies(chilly)$.

Minimal models and CWA/CGWA

Circumscription need not produce a unique interpretation

Suppose KB = {… , $Bird(c)$, $Bird(d)$, $(\neg Flies(c) \vee \neg Flies(d))$}

Because we need to consider what is true **in all minimal models**, we see that

$KB \not\models_{\leq} Flies(c)$ and $KB \not\models_{\leq} Flies(d)$

In other words, we cannot conclude that c is a normal bird, nor that d is, but only that one of them is normal: $KB \models_{\leq} Flies(c) \vee Flies(d)$

With CWA we would add the literal $\neg Ab(c)$, and by similar reasoning $\neg Ab(d)$, leading to inconsistency.

Thus circumscription is more cautious than the CWA in the assumptions it makes about "controversial" individuals, like c and d.

The GCWA would not conclude anything about either the denotation of c or d.

Circumscription and quantified sentences

Circumscription works equally well with unnamed individuals. Suppose:

$$\exists x\, [Bird(x) \land (x \neq chilly) \land (x \neq tweety) \land InTree(x)]$$

We can conclude:

$$\exists x\, [Bird(x) \land (x \neq chilly) \land (x \neq tweety) \land InTree(x) \land Flies(x)]$$

In the minimal models there will be a single abnormal individual, Chilly. If:

$$\exists x\, [Bird(x) \land (x \neq chilly) \land (x \neq tweety) \land \neg Flies(x)]$$

a minimal model will have exactly two abnormal individuals.

Open issues with circumscription

Although the default assumptions made by circumscription are usually weaker than those of the CWA, there are cases where they appear too strong.

Suppose, for example, that we have the following KB:

$$\forall x\, [Bird(x) \land \neg Ab(x) \Rightarrow Flies(x)]$$

$$Bird(tweety)$$

$$\forall x\, [Penguin(x) \Rightarrow (Bird(x) \land \neg Flies(x))]$$

From this follows:

$$\forall x\, [Penguin(x) \Rightarrow Ab(x)]$$

Minimizing abnormalities leads to:

$$KB \models_{\leq} \neg \exists x\, Ab(x)$$

$$KB \models_{\leq} \neg\, Penguin(tweety)$$

$KB \models_{\leq} \neg \exists x\, Penguin(x)$ i.e. there are no penguins, too strong

Partial fix

McCarthy's definition related to *predicate completion*. Let **P** and **Q** be sets of predicates

Ordering on interpretations:

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

1. for every $P \in \mathbf{P}$ $\mathrm{I}_1[P] \subseteq \mathrm{I}_2[P]$ holds — **P** variable predicates
2. for every $Q \in \mathbf{Q}$ $\mathrm{I}_1[P] = \mathrm{I}_2[P]$ holds — **Q** fixed predicates

so only predicates in **P** are allowed to be minimized.

Previous example: **P** = $\{Ab\}$ and **Q** = $\{Penguin\}$; minimize Ab, keeping $Penguin$ fixed.

Problems:

- need to decide what to allow to vary: what about $Flies$?
- cannot conclude $\neg Penguin(tweety)$ by default!
  (only get default $(\neg Penguin(tweety) \Rightarrow Flies(tweety))$)

Default logic [Reiter]

Beliefs as deductive theory

- *explicit beliefs* = axioms
- *implicit beliefs* = theorems = least set closed under inference rules

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

- F is a set of sentences (facts)
- D is a set of **default rules**: $\frac{\alpha : \beta}{\gamma}$

  read as “If you can infer α, and *it is consistent to assume* β, then infer γ”

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

e.g. $\frac{Bird(tweety) : Flies(tweety)}{Flies(tweety)}$ also $\frac{Bird(x) : Flies(x)}{Flies(x)}$

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

Extensions

Problem: how to characterize theorems/entailments

- cannot write a derivation, since do not know when to apply default rules
- no guarantee of unique set of theorems

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

E is an **extension** of $\langle F, D \rangle$ iff for every sentence π, E satisfies the following:

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

So, an extension E is the set of entailments of $F \cup \{\gamma\}$, where the γ are a "suitable" set of assumptions given D. Note that α has to be in E, not in F. This has the effect of allowing the prerequisite to be believed as the result of other default assumptions.

Note that this definition is *not constructive*.

Example: single extension

Suppose KB is:

$$F = \{Bird(chilly), \neg Flies(chilly), Bird(tweety)\}$$

$$D = \{\frac{Bird(x) : Flies(x)}{Flies(x)}\}$$

then there is a unique extension, where $\Delta = \{Flies(tweety)\}$

- This is an extension since $Bird(tweety) \in E$ and $\neg Flies(tweety) \notin E$.
- No other extension, since $Flies(tweety)$ in any estension and no extension has $Flies(chilly)$.

If E is *inconsistent* we can conclude anything we want.

Theorem: An extension of a default theory is inconsistent *iff* the original F is inconsistent. In this case the extension is unique.

But in general a default theory can have **multiple extensions**.

Example: multiple extensions

The **Nixon diamond**:

$F = \{Quacker(nixon), Republican(nixon)\}$

$D = \{Quacker(x) : Pacifist(x) \ / \ Pacifist(x),$

$Republican(x) : \neg Pacifist(x) \ / \ \neg Pacifist(x)\}$

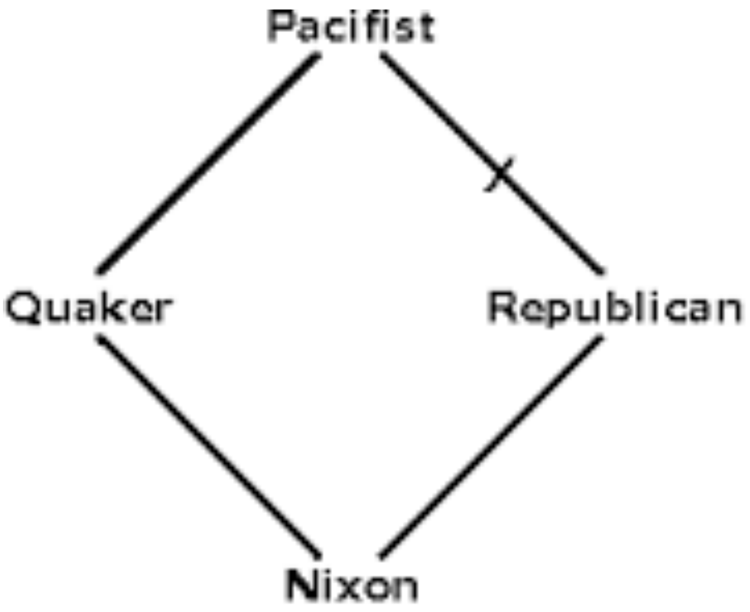
Two extensions:

E_1 has Δ = $\neg Pacifist(nixon)$

E_2 has Δ = $Pacifist(nixon)$

Which to believe? Two possible approaches:

1. **credulous**: choose an extension arbitrarily
2. **skeptical**: believe only what is common to all extensions



Properties

1. If a default theory has distinct extensions, they are **mutually inconsistent**.

   $F = \{A \vee B\}$ $D = \{:\neg A/\neg A, :\neg B/\neg B\}$ $E_1 = \{A \vee B, \neg A\}$ $E_2 = \{A \vee B, \neg B\}$

2. There are default theories with **no extensions.**

   Consider the default: $:A/\neg A$. If $F = \{\ \}$ then $E = \{\ \}$

2. Any **normal** default theory has an extension.

3. Adding new normal default rules does not require the withdrawal of beliefs, even if adding new beliefs might. Normal default theories are **semi-nonmonotonic**.

Grounded extensions

We have a problem that leads to a more complex definition of extension.

Suppose $F = \{\ \}$ and $D = \{: p\ / p\}$

Then $E =$ entailments of $\{p\}$ is an extension since $p \in E$ and $\neg p \notin E$.

However, we have no good reason to believe p ! Only support for p is the default rule, which requires p itself as a prerequisite. So the default should have no effect.

Desirable extension is only: $E =$ entailments of { }

A revision of the definition of extension is necessary. Reiter's definition:

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

$$\text{if } \alpha : \beta\ /\ \gamma \in D,\ \alpha \in \Gamma(S),\ \text{and } \boldsymbol{\neg\beta \notin S},\ \text{then } \gamma \in \Gamma(S) \quad [\text{instead of } \neg\beta \notin \Delta(S)\ ]$$

A set E is an extension of $\langle F, D\rangle$ iff $\boldsymbol{E = \Gamma(E)}$, **i.e.** E is a fixed point of the Γ operator.

Reason Maintenance Systems

TMS, ATMS

Belief revision

Many of the inferences drawn by a knowledge representation system will have only **default** status or **tentative** nature. Inevitably, some of these inferred facts will have to be retracted.

This process is called **belief revision**.

Suppose: $\{P,\ P \Rightarrow Q\} \in KB$ and $\text{T}\small{\text{ELL}}(KB, \neg P)$.

How to avoid a contradiction? $\text{R}\small{\text{ETRACT}}(KB, P)$? What about inferred facts such as Q?

Suppose: $\{P,\ P \Rightarrow Q, R, R \Rightarrow Q\} \in KB.$ What about Q?

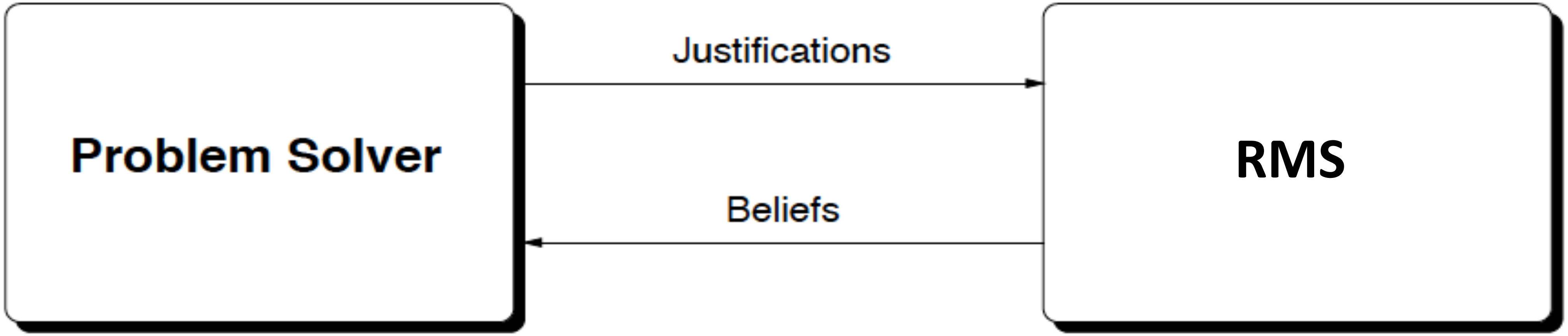
One simple approach to belief revision is to number sentences according to the order of assertion: $P_1, P_2 \ldots P_n$.

When the call $\text{R}\small{\text{ETRACT}}(KB, P_i)$ is made, the system reverts to the state just before P_i was added, removing both P_i and any inferences derived from P_i.

Sentences P_{i+1} through P_n can then be added again if it is the case.

Reason maintenance systems (JTMS, ATMS), are reasoning mechanisms designed to handle these problems efficiently.

Architecture



- The problem solver communicates facts, rules, assumptions along with their **justifications** to the RMS. It may retract assertions.
- The RMS maintains beliefs, detects contradictions, performs beliefs revision, generates explanations and provides the PS current beliefs.

Reason Maintenance Systems

Rational thought is the process of finding reasons for attitudes. (Doyle 1979]: justified belief or reasoned argument, rather than truth.

The RMS handles nodes as **propositional variables**, representing propositions, rules and justifications.

Nodes are of different **types**: premises, assumptions, contradictions and have different **support** according to the type of RMS.

We will look at two of them:

1. JTMS (**Justification-based** Truth Maintenance Systems)

   John Doyle "A Truth Maintenance System", *Artificial Intelligence* 12:231-272, 1979.

2. ATMS (**Assumption-based** Truth Maintenance Systems)

   Johan de Kleer, "An assumption-based TMS", *Artificial Intelligence*, 28:127–162, 1986.

JTMS: adding justifications

JTMS: **Justification-based Truth Maintenance System**

Each assertion in the knowledge base is represented as a node in the TMS annotated with a **justification** consisting of the set of sentences from which it was inferred.

$\{P \Rightarrow Q\} \in KB\,;\ \textsc{Tell}(KB, P)$

will cause Q to be added as node with justification $\{P,\ P \Rightarrow Q\}$

Justifications can be more than one set.

Justifications can be used to make retraction efficient …

Maintenance with justifications

With JTMS, $\textsc{Retract}(KB, P)$ will delete exactly those sentences for which P is a member of every justification.

- If a sentence Q had the single justification $\{P, P \Rightarrow Q\}$, then Q would be removed;
- if it had the additional justification $\{P, P \vee R \Rightarrow Q\}$, then Q would still be removed;
- if it also had the justification $\{R, P \vee R \Rightarrow Q\}$, then Q would be spared.

Ins and outs:

The JTMS, rather than deleting a sentence from the knowledge base entirely when it loses all justifications, **marks the sentence as being "out"**.

If a subsequent assertion restores one of the justifications, then the sentence is marked as "**in**" again. No need to re-compute inferences.

Justifications

Proposition P ca be in one of two states. Either

a) P has at least one currently acceptable (valid) reason, or

b) P has no currently acceptable reasons (either no reasons at all, or only unacceptable ones).

If P falls in state a), we say that P is **in** the *current set of beliefs*, otherwise that P is **out**.

A **reason** (or [**nonmonotonic] justification**) for a belief is a pair of sets of beliefs (*inlist*, *outlist*), the set of propositions that should be in (or out) for the beliefs to be in.

A node can have more than one justification in its **support list** (SL).

Examples:

R	SL: $(\{\ \}\ \{\ \})$	**premise**, empty support lists, always **valid**
Q	SL: $(\{P, P \Rightarrow Q\}, \{\})$	normal inference, valid if *inlist* elements are **in**
P	SL: $(\{\ \}\ \{\neg P\})$	**assumption** because *outlist* not empty, valid if *outlist* elements are **out**

Example

	Propositions	Justifications	Context In	Context Out
	A: Temperature >= 25	({ }, {B}) assumption	A	B
	B: Temperature < 25			
	C: Not raining	({ }, {D}) assumption	A, C	B, D
	D: Raining			
	E: Day	({ }, {F}) assumption	A, C, E	B, D, F
	F: Night			
PS ⇒	G: Nice weather	({A, C}, { })	A, C, E, G	B, D, F
PS ⇒	H: Swim	({E, G}, { })	A, C, E, G, H	B, D, F
PS ⇒	I: Contradiction	({C}, { }) backtracking		
	X: Handle	({ }, { }) premise		
	D: Raining	({X}, { })	A, D, E	B, C, F, D, H, I
PS ⇒	J: Read	({D, E}, { })	A, D, E, J	B, C, F, D, H, I

Hypothetical reasoning with JTMS

JTMSs can be used to speed up the analysis of **multiple hypothetical situations.**

Example: 2048 Olympic Games in Romania. Which sports in which towns?

$Site(Swimming, Pitesti), Site(Athletics, Bucharest), Site(Equestrian, Arad)$

Compute all the consequences.

Now try $Site(Athletics, Sibiu)$, the JTMS takes care of all the revisions.

In a JTMS, the maintenance of justifications allows you to move quickly from one state to another by making a few retractions and assertions, but at any time **only one contexts is represented**.

Taking the idea one step forward we could let multiple contexts to co-exhist.

Assumption-based Truth Maintenance Systems

An **Assumption based Truth Maintenance System** (ATMS) represents all the contexts been considered *at the same time*. Alternative contexts are explicitly stored.

An **ATMS-node** is characterized by a *label* and *justifications*

1. In an ATMS each sentence/node maintains a **label** consisting in a number of assumption sets (**environments**).
2. **ATMS justifications** are Horn formulas of the form:

$$L_1, L_2, \ldots\ L_n \rightarrow C$$

where L_1, L_3, …, L_n are the antecedents, and C is the consequent of justification, corresponding to the node being justified.

ATMS

Three types of nodes:

- **Premise nodes**. Always true. With label {{ }}. True in every consistent environment.
- **Assumption nodes**. Assumptions are never retracted.
- **Contradictions**. Every environment which allows a contradiction is **inconsistent**. These environments are called **nogoods.**

The fundamental operation is deciding whether a proposition **holds in a given environment:**

> A node *n* **holds** in a given environment *E*, iff it can be derived from *E* given the set of justifications *J*: $E, J \vdash n$.

ATMS and explanations

ATMS allow for quickly generating **explanations**:

An explanation of a sentence P is a set of sentences E such that $E \vDash P$, *usually we prefer a minimal one*

Explanations can only be assumptions: e.g. different causes for the car not starting (battery dead, no gas in car …).

ATMS can generate explanations by making, even if some assumptions are contradictory. The label for the sentence “car won’t start” contains the assumptions that would justify the sentence

Conclusions

We have seen three ways of deal with defaults:

1. CWA: we try to complete the KB adding negative facts and use normal entailment. Nice computational properties for subsets of FOL.
2. Circumscription: tries to restrict the possible interpretations to the minimal ones, for certain predicates that we define as “*abnormal*”.
3. Default logic: tries to characterize a new form of *tentative inference* through default rules and the notion of extensions.
4. TMS, ATMS are computational mechanisms to support defeasible reasoning and belief revision.

Your turn

- Theory of belief revision (AGM postulates).
- Algorithms for TMS.
- Algorithms for ATMS.

References

Knowledge Representation and Reasoning, Ronald Brachman and Hector Levesque, Elsevier, 2004 [Ch. 11]

Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach* (3rd edition). Pearson Education 2010 [Ch. 12]

Genesereth, M., and Nilsson, N., Logical Foundations of Artificial Intelligence, San Francisco: Morgan Kaufmann, 1987 [Ch. 6]