Default Logic
Autoepistemic Logic

Non-classical logics and application seminar, winter 2008

Mintz Yuval

Introduction and Motivation

- Birds Fly…
- As before, we are troubled with formalization of Non-absolute sentences.
- Classical logic deals with absolutes – can't capture the essence of "most" or "usually".
- We therefore turn to non-monotonic reasoning.
- 2 distinct directions for formalization of such sentences will be given – Default logic and Autoepistemic logic.

Introduction – Default logic

- “Birds fly, and tweety is a bird”
- when can we assume tweety flies?
- given no evidence to the contrary, we should believe tweety flies
- We’ll split our theory to certain and uncertain things, and deal differently with each.

Default Logic Introduction - continued

- Similarity exists to the closed-world assumption- both are mechanisms to add facts.
- But not too similar:
  - we will be adding positive literals as well.
  - “Tweety flies” is an example of such.
- we’ll create rules that will allow us to extend our theory.
- In this presentation - first order default logic.

Introduction– Autoepistemic logic

- Why should we split our theory?
- We would like to be able to reason about every part of our theory.
- The Contraptive Example:
  - chilly is a non-flying animal, and usually birds fly. (birds are animals, although its not required information).
  - we would like to be able to infer that chilly is probably not a bird.

Autoepistemic Introduction – continued

- We'll use modal epistemic operators of Belief to formalize all of our sentences.
- Our story "tweety is a bird" and "if tweety is a bird and we don't believe it can't fly, then it flies" will both be valid sentences in our theory.
- Since we'll be using epistemic logic to talk about our own set of beliefs, it will be called Autoepistemic.

Default Theory

- A default theory is a pair <D, F>, where:
  - F is a set of closed formulae, called ‘Facts’.
  - D is a set of default rules.
- F consists of All facts that are known in the classical sense (absolute).
- D will contain the mechanism by which we’ll extend our theory.

Default Rules

- Rules that take the following form:

$$\frac{\alpha(x) : \beta_1(x), \beta_2(x), \ldots, \beta_m(x)}{w(x)}$$

- $\alpha(x)$, $\beta_i(x)$ and $w(x)$ are sentences whose free variables are of x:
  - $\alpha(x)$ is the ***precondition*** (or prerequisite).
  - $\beta_i(x)$ are ***justifications.***
  - $w(x)$ is the consquent (or ***conclusion***).
- Can also be written as $\langle\alpha(x) : \beta(x) / w(x)\rangle$ (we'll use this form of writing)

Default Rules, continued

- We would say that the conclusion is achieved when
  - the precondition is inferred from the theory.
  - all the justifications are consistent with our theory.
- Since the theorem we would obtain would be a first order logic, it would be sound and complete.
- Therefore, we can use the alternative semantic notion. (instead of syntactic).

Instance of a default rule

- An **instance** of a default rule is obtained by uniformly substituting ground terms for the free variables in the default.
- Example: consider our usual "Tweety is a bird, and birds fly" theory.
- our default rule would probably be –
  - <Bird(x):Fly(x)/Fly(x)>
- Therefore, an instance of it would be –
  - <Bird(tweety):Fly(tweety)/Fly(tweety)>

Normal and Semi-normal rules

- a **normal rule** is a rule of the form
  - $< \alpha(x) : \beta(x) \,/\, \beta(x) >$
  - a special case – when there's no precondition. we get a rule of the form $< : \beta(x) \,/\, \beta(x) >$
- a **semi-normal rule** is a rule of the form
  - $< \alpha(x) : \beta(x) \wedge w(x) / w(x) >$

Default Extension – definition

- Given a default theory T = <F,D>, we would say that a set of sentences ε is an extension to the theory if and only if For each sentence π, π⨂ε if and only if F⇪🖓 π, where

  🖓= {♎ | <♋:♌/♎> ⨂ D , ♋ ⨂ ε, ¬ ♌ ⫝ ε}

- It's clear that F ⊘ ε, since for sentence π⨂F, clearly F π

A simple example – back to birds

- Consider the following default theory, in which:
  - F = {Bird(tweety), Bird(chilly), ¬Flies(chilly) }
  - D = {<bird(x): flies(x) / flies(x) >}
- F∪**{**flies(tweety)**}** is a possible extension.
- Our extension could never include flies(chilly):
  - It requires an instance of <Bird(chilly):flies(chilly)/flies(chilly)>
  - But ¬Flies(chilly) is in our facts, so we could never consistenctly fulfill its justification.

Default extension- regarding consistency

- We wish our extension to be consistent.
- If F is consistent, and all our default rules are either normal or semi-normal, than every extension we can create for the theory will be consistent.
  - Consider a default rule <:x/y>
- Of course, if F is inconsistent our extension will be inconsistent.

Explanation definition

- If g is a closed formula, E is an **Explanation** of g from <D,F> if E is the set of consequents of some D’, a set of instances of elements of D such that:
  - $E \cup F \models g$
  - $E \cup F$ entails the precondition of D’.
  - All justification of D’ are consistent with some extension of <D,F> that contain E.

How explanation helps

- Gives us “the glue” that connects expansions and first-order logic proofs.
- A formal minimal notion to infer a sentence from our default theory.
- we don’t require an entire expansion for a proof. sometimes proving that a sentence exists in some extesion (More on this later) is easier than finding the extesion.

multiple extensions?

- consider the following default theory:
  - F = {Republican(dick), Quaker(dick)}
  - D = { <Republican(x):¬pacifist(x)/ ¬pacifist(x) >,
    <Quaker(x): pacifist(x)/ pacifist(x)> }
- How can we extend this theory?

Multiple extensions

- pacifist(dick) will be in a valid expansion.
  - It can be explained by {pacifist(dick)}, which is the consequent of <quacker(dick):pacifist(dick)/pacifist(dick)>
- But ¬pacifist(dick) is also in a valid expansion!
  - It can be exaplined by {¬ pacifist(dick)} , which is the consequent of <republican(dick):¬pacifist(dick)/¬pacifist(dick)>

The skeptical reasoner vs. the brave (credulous) reasoner

- we sometimes reach a situation in which several default rules will allow us to reach several different extensions.
- 2 immediate attitudes are possible:
  - The **skeptical** reasoner will believe only in sentences common to all extensions.
  - The **brave** reasoner will choose one extension of the default theory as a basis set of setences.

A need for a slightly different definition

- Our definition for an extension was not very constructive.
- Consider the following basis:
  - “those who eat onion soup eat onions”
  - “those who eat onion soup love eating”
  - “those who love eating brush their teeth”
  - “those who brush their teeth care for their personal hygiene”
  - “those who care their personal hygiene don’t eat onions”
  - “Yuval eats onion soup”
- We can have 2 possible extensions – in one we can explain that yuval eats onions, and in the other the opposite.
- But do we truly consider both as likely?

Default extension – iterative definition

- Given a default theory $<D,F>$, we'll consider a sequence of formulae sets s_0, s_1…, $S = \cup\ s_i$, $s_0 = F$ and:
  - $S_{i+1} = s_i \cup \{w(c) \mid <\alpha(c) : \beta(c) / w(c)>$ is an instance of a default from D
  - $\alpha(c)$ follows from s_i
  - $\beta(c)$ is consistent with S for all $\beta i(c)$ ($\beta(c) = \beta 1(c)$,…, $\beta n(c)$)
- the set of consequents in S will be called an extension.

Back to our example

- Notice we require justification to be consistent with S (as opposed to Si)
  - It might otherwise have prevented multiple extensions.
  - More problematic – it could have caused inconsistency.
- If we'll look back at the example, we now may have the basis to claim one extension as more likely to happen.

The art of creating default rules

- Big Issue with default logic – extensions are subject to the exact way we formalized our rules.
- Since we can't reason about default rules, we sometimes can't prove things we would expect to be able to.
- For example "Birds fly and fred doesn't fly" – it's likely that fred is not a bird.
- if we'll formalize this as we did before (all the regualr tweety examples)– we wouldn't be able to prove it.

Normal Rules with no precondition

- However, we could 'manipulate' things:
- Consider <D,F> where
  - D={<:BirdFly(x) / BirdFly(x)>}
  - F={for each x,Birdfly(x)∧bird(x)→flies(x), ¬flies(fred) }
- We can explain ¬bird(fred) using
  F∪{birdsfly(fred)}.
- However, this is not very attainable solution (in general).

Semi-normal rules, problems with disjunction

- If we thought Fred had a problem, what about situations in which we can prove things we didn't intend to?
- Like we've seen in regard to cwa, when our facts contain disjunctions, we might find ourselves with problematic conclusions.
- Using normal rules will save us the problem (or most of it), since it adds all of the justifications as conclusions.
- But what about semi-normal rules?

Semi-normal rules, problems with disjunction

- Consider <F,D> where
  - D={<bird(x):flies(x) $\wedge\neg$baby(x)/flies(x)>}
  - F ={bird(pete), bird(mary), baby(pete)$\vee$baby(mary) }
- We can explain flies(pete) $\wedge$ flies(mary) (although we know for certain one of them at least is a baby).
- Notice this is not a problem in consistency, as F doesn't contain any explicit rule that connects babies and flight ability.
- What it does show is that our formalization is lacking- we never intended for this to be valid.

Semi-normal rules, more problems

- Consider the following theory:
  - D = {<bird(x):flies(x) $\wedge$ ¬baby(x)/ flies(x)>,
    <bird(x):cries(x) $\wedge$ baby(x)/ cries(x)>}
  - F = {bird(tweety)}
- This default theory will allow us to explain flies(tweety) $\wedge$ cries(tweety).
- It's based on the fact we can consistently add both baby(x) and ¬ baby(x) (separately).
- Again – this is probably not what we've intended.

When does an extension exist?

- Consider the following:
  - $D = \{\langle a:b \wedge c/c\rangle, \langle c:\neg b/\neg b\rangle \}, F = \{a\}$
- This theory has no extensions.
- the only cases where there are no extension is when there exists circularity.
- Circularity - defaults in which the justification of one is inconsistent with the consequent of the other, which must be subsequently applied.
- **Ordered** default theories disallow such circularity.
- Ordered semi-normal default theories like this will always have an extension.

A few words about equality

- Default theory can be used with first-order logic with equality as well.
- Our default rules could then include statements of equality or inequality.
- Default rules can be used to derive inequalities:
  - D={<:p(x)/p(x)>}, F={¬p(A)}.
  - We can conclude p(B), from which it logically follows that A != B

A few more words about equality

- Unique name assumption –
  - consider the rule <:x != y/ x != y>
  - this is an embodiment of the unique name assumption as a default.
  - In the same manner < :¬x/¬x> is in fact an embodiment of cwa.
- As expected, can be used to imply equality -
  - <P(x) = P(y) : x = y / x = y >

Stable model semantics

$$\frac{\alpha_1 \wedge \ldots \wedge \alpha_n : \neg\alpha_{n+1}, \ldots, \neg\alpha_m}{\alpha_0}$$

- A special case, in which:
  - F consists of a conjunction of atoms.
  - Consequents of defaults are atoms.
  - Justifications of defaults are negations of atoms.
  - Preconditions are conjunctions of atoms.
- our default theory define the same behaviour as the Prolog program, with negation as failure.

Forward and backward chaining

- 2 ways of trying to implement default reasoning and create an extension.
- **Forward chaining** – simply run, choose defaults whose precondition is derived. Rinse and repeat.
- **Backward chaining** – starting from assumed conclusions, we try to determine if formula can be consistently explained via all justifications in instantiations of the default rules.

Complexity

- Unsurprisingly, default logic problems are very hard to implement.
- For a default theory for propositional logic, determining if a proposition can be explained by the theory is decidable, but NP-complete.
- For first-order logic, it's not even semi-decidable.
- On weakened logics some aspects can be determined in polynomial time.

More about complexity

- Finding an extension for an ordered, disjunction free, unary defaults- can be done in an $O(n^2)$ algorithm.
  - the general version of this problem is NP-complete
- For a Horn default theory, there's an $O(n)$ algorithm for finding whether a certain literal exists in any extension.
  - the general version of this problem is NP-hard, even for disjunction free unary defaults
- For a Horn default theory, there's an $O(n^3)$ algorithm for finding whether a certain literal exists in all extensions.
  - the general version of this problem is co-NP-hard

From Default logic...

- Default logic was non-monotonic due to its being **defeasable**.
- When given information for our theory such as bird(tweety), we’ve found it likely to assume that tweety flies.
- We found it likely to assume that its true - but it might have been wrong.
- If we’ll look solely on our facts, we can find a model that satisfies all of them, yet doesn’t satisfy our conclusion

...To Autoepistemic logic

- In Autoepistemic logic, we will reason about our set of beliefs.
- “all birds that can be consistently asserted to be capable of flight are capable of flight”.
- Earlier, we’ve formalized this with default rules.
- But if we are capable of reasoning about our beliefs, we’ll be able to formalize this rule completely within our theory.

Autoepistemic, continued

- Autoepistemic logic is non-monotonic due to the fact its **indexical**.
- Consider the last statement about birds.
- It means that the only birds who can't fly are those that were explicitly mentioned as not capable.
- Therefore, given tweety is a bird, and we didn't assert its inability to fly – it MUST fly.
- Our proofs deal mainly with propositional logic, since there are issues with quantifying into a modal operator scope.

The consistency Operator

- This will be our dual modal operator.
- We will right it as **M**.
- **M**α means α can be consistently asserted.
- Informally, the inference we would like to give the consistency operator is – "**M**α is derivable if α isn't derivable".
- Remember the "Unless" operator?

The Belief Operator

- Our main modal operator of belief.
- We'll write it as **B**.
  - Also referred to as L in the literature.
- To say **B**α will mean (informally) that we believe in α.
- The dualism between consistency and belief –
  - **B** == ¬**M**¬
- Since the fundamental notion of this logic is to formalize beliefs, it was chosen as the main operator.

A simple example

- Consider the following theory:
  - Bird(tweety)
  - Bird(twetty) ∧ ¬**B**(¬can-fly(tweety)) → can-fly(tweety)
- We would like to reach a formalization in which every model that satisfies this theory will satisfy the conclusion can-fly(tweety).
- If we would add ¬can-fly(tweety), we would have a different theory – in which we will never expect to reach this conclusion.

Autoepistemic theory

- A simple propositional logic theory, with the addition of the Belief operator in its formulae.
- Represents the total belief of a rational agent reflecting on his beliefs.
- To determine an Autoepistemic theory, we need to determine 2 things:
  - Which propositional constants are true in the real world. These constants Contain no **B** operators (**objective formulae**)
  - Which formulae the agent (we) believe. **B**α is true only if α is in the agent set of beliefs.

Propositional Interpretation

- first stage in defining a model for an Autoepistemic theory T.
- We assign truth values to all formulae of the language of T.
- This assignment should be consistent with truth-recursion of propositional logic.
- We assign **arbitrary** truth values to all constants and formulae of the form **B**α.

Propositional model

- A **propositional model** of an Autoepistemic theory T is a propositional interpretation of T in which all formulae of T are true.
- Propositional model inherit propositional logic soundness and completeness theorem.
- Therefore – a formula P is true in all propositional models of an Autoepistemic theory T iff it can be derived from T using usual rules for propositional logic.

Autoepistemic Interpretation

- An **Autoepistemic Interpretation** of an Autoepistemic theory T is a propositional Interpretation of T in which **B**α is true iff α is true.
- An **Autoepistemic Model** of an Autoepistemic theory T is an Autoepistemic interpretation of T in which all formulae of T are true.

Definition via previous example

- Consider our previous example:
  - Bird(tweety)
  - Bird(twetty) ∧ ¬**B**(¬can-fly(tweety)) → can-fly(tweety)

	Bird(tweety)	Can-fly(tweety)	**B**(¬can-fly(tweety))
Propositional interpretation	F	T	F
Propositional model	T	F	T
Autoepistemic interpretation	T	F	F
Autoepistemic model	T	T	T

The problem of inference in non-monotonic logic

- We now have a formal notation of semantics for Autoepistemic theory.
- But what about syntactic notation?
- Monotonic logic's inference rules are monotonic themselves.
- That allows us to try and infer in an iterative process.
- In non-monotonic logic, that is not so.

Competence Model

- We won’t be actually giving a syntactic notation.
- Instead, we’ll describe a “competence model” – Autoepistemic theory that capture every belief we can conclude.
- Those theories will be sound and complete.

Soundness

- We would say an Autoepistemic theory T is sound with respect to an initial set of premises A iff:
  - Every Autoepistemic interpretation of T which is a propositional model of A is a model of T.
- Intuitively – if all our premises are true, then our theory is true as well.

Semantic Completeness

- We would call an Autoepistemic theory T semantically complete, iff:
  - T contains every formula that is true in every autoepistemic model of T.
- Intuitively, if a formula is true under every autoepistemic model of an agent, it means it must be true whenever all the agent's beliefs are true.
- Since the Agent is rational – he should be able to infer that.

Stable Autoepistemic Theory

- Given an autoepistemic theory T, we'll require 3 things from it for us to call it '**stable**':
  - if $P_1,\ldots,P_n$ are in T, and $P_1,\ldots,P_n \vdash Q$, then Q is in T (ordinary tautological consequence).
  - Positive Introspection – if $\alpha \in T$ then $B\alpha \in T$.
  - Negative Introspection – if $\alpha \notin T$ then $B\alpha \notin T$.
- Stable "in the sense that no further conclusions could be drawn by an ideal rational agent in such a state"

Stable Autoepistemic Theory

- If we have a stable autoepistemic theory which is also consistent, then it will satisfy 2 more conditions:
  - if $B\alpha \in T$ then $\alpha \in T$.
  - if $B\alpha \notin T$ then $\alpha \notin T$.
- The stable autoepistemic theories will assure us that our theory is semantically complete

Grounded Autoepistemic theories

- We will say that an autoepistemic theory T is grounded in a set of premises A if:
  - Every forumla of T is included in the tautological consequence of $A \cup B1 \cup B2$.
  - $B1 = \{B\alpha \mid \alpha \in T\}$
  - $B2 = \{\neg B\alpha \mid \alpha \notin T\}$
- T will be sound with regard to a set of premises A iff T is grounded in A.

Expansions

- We've seen that when given a set of premises A, a rational agent could be expected to believe the a stable autoepistemic theory grounded in A.
- We call this "stable expansions of A"
- There can cases in which more than 1 stable expansion possible:
  - Consider $A = \{\neg \mathbf{B}\alpha \rightarrow \beta, \neg \mathbf{B}\beta \rightarrow \alpha\}$

Expanions, continued

- There can be cases where no expansions is possible:
  - Consider A = {¬**B**α→ α }
- Sometimes, we can get theories to which are definition is lacking.
  - Consider A = {**B**α→ α}
  - We have 2 possible expansions, but why should we believe in α?

Enumerating stable expansions

- Given as a constructive way to try and find a model for a given Autoepistemic theory T. 4 simple steps:
  1. Replace every $B\alpha_i$ with either True or False.
  2. We now have a propositional theory. We'll simplify it and call it T'. If it isn't consistent, we have a bad assignment.
  3. For each α_i if $B\alpha_i$ that was given the value True, confirm that T' satisfies α_i. For each that was given value false, confirm that T' doesn't satisfy α_i.
  4. If 3 was true for all i's, then T'`s entaliments form the objective part of a stable expansions (And the non objective part can be added appropriately).

Enumerating - problems

- quite a problematic solution.
- exponential in the number of expressions containing belief operators
  - we need to check every possible combination of assignments of true and false to them.
- And that's in the case of propositional logic, in which checking satisfaction can be done in reasonable time.

Enumerating - example

- Consider the propositional case of the bird problem.
- Our theory T contains the following:
  - T = {Bird(Tweety), Bird(chilly), ¬flies(chilly),
    bird(tweety) ∧ ¬**B**(¬ flies(tweety)) → flies(tweety),
    bird(chilly) ∧ ¬**B**(¬ flies(chilly)) → flies(chilly) }
- We have 4 assignments to check, since **B**(¬flies(chilly)) and **B**(¬flies(tweety)) can receive truth assignments independently.

Enumerating – example, continue

- **B**(¬flies(chilly)) true **B**(¬flies(tweety)) true.
  - After simplification, our theory T’ = T. therefore, ¬flies(tweety) is not entailed from T’, and this assignment is wrong.
- **B**(¬flies(chilly)) false **B**(¬flies(tweety)) true.
  - After simplification, our theory T’ = T ∪ {flies(chilly)} – it’s inconsistent.
- **B**(¬flies(chilly)) true **B**(¬flies(tweety)) false.
  - After simplifcation, T’ = T ∪ {flies(tweety)}. This is a valid assignment, as ¬flies(chilly) is entailed by it, and ¬flies(tweety) is not. Therefore, T’ can be a basis for stable Autoepistemic expansion.

Correlation between Autoepistemic and Default logic

- we look at a form of ‘strongly grounded’ Autoepistemic logic, in which all formulae are of form:
  - $B\alpha \wedge \neg B \neg \beta_1 \wedge \ldots \wedge \neg B \neg \beta_m \rightarrow w$
- Formula like this can be interpreted as the default rule $<\alpha:\beta_1,\ldots, \beta_m/w>$

Conclusion

- As we’ve just seen Autoepistemic logic is more “expressive” than default logic.
- As such, it is also more abstract.
- Both have many variants – and still a question remains on how to correctly model a given theory, as all ‘fail’ on specific pathological cases.