Professional Skills in Computer Science
Lecture 11: Deduction and Argumentation

Ullrich Hustadt

Department of Computer Science
School of Electrical Engineering, Electronics, and Computer Science
University of Liverpool

Deduction Rules: Chain Argument

Chain Argument aka Law of syllogism:
– A implies B
– B implies C
– Therefore, A implies C

Example:
– If Ben is sick, then he cannot study
– If Ben cannot study, then he will perform badly in the exam
– If Ben is sick, then he will perform badly in the exam

Ben should hand in a claim of mitigating circumstances!

Beliefs and Time

Example:
– If Ben is sick, then he cannot study
– If Ben cannot study, then he will perform badly in the exam
– If Ben is sick, then he will perform badly in the exam
– Ben should hand in a claim of mitigating circumstances!

This is surprisingly difficult to formalise:

- Ben must hand in his claim before he knows his exam result, in particular, at least one week before the meeting of the Board of Examiners that considers the exam results
  - ⇝ formalisation has to talk about time
  - ⇝ formalisation requires temporal logic
- Therefore, Ben must hand in a claim if he believes that he has (or will) be performing badly in the exam
  - ⇝ formalisation has to talk about beliefs
  - ⇝ formalisation requires doxastic logic

Beliefs and Time

Example:
– If Ben is sick, then he cannot study
– If Ben cannot study, then he will perform badly in the exam
– If Ben is sick, then he will perform badly in the exam
– Ben should hand in a claim of mitigating circumstances!

Formalisation requires

- temporal logic
  - ⇝ covered in COMP313 Formal Methods
- doxastic logic
  - ⇝ covered in COMP304 Knowledge Representation and Reasoning

Equality

– $A(o_1)$
– $o_1 = o_2$ (o_1 and o_2 be equal)
– Therefore, $A(o_2)$

Examples:
– 2 is a prime number
– $2 = \sqrt{4}$
– Therefore, $\sqrt{4}$ is a prime number

– Batman is a superhero
– Bruce Wayne is Batman
– Therefore, Bruce Wayne is a superhero

– Ben believes that Batman is a superhero
– Bruce Wayne is Batman
– Therefore, Ben believes that Bruce Wayne is a superhero

The last two examples cause philosophers quite a bit of concern

Denying the antecedent

Consider the following deductive argument:
– If the police knew that Ben had a motive to kill Eve, then he would be a suspect
– The police do not know that Ben had a motive
– Therefore, Ben is not a suspect

This is not a valid argument!
Ben may still be a suspect, but for other reasons, for example, there was an eye witness

- The above is an example for the fallacy of denying the antecedent, an invalid form of deductive reasoning
  – A implies B
  – Not A
  – Therefore, not B
- Remember: Denying the consequent is a valid form of deductive reasoning

Affirming the consequent

Consider the following deductive argument:
– If the police knew that Ben had a motive to kill Eve, then he would be a suspect
– Ben is a suspect
– Therefore, the police know that Ben had a motive

This is not a valid argument!
Just as before, Ben may be a suspect for other reasons and it is possible that the Police have no clue about his motive

- The above is an example of the fallacy of affirming the consequent, an invalid form of deductive reasoning
  – A implies B
  – B
  – Therefore, A
- Remember: Affirming the antecedent is a valid form of deductive reasoning

Faster than light travel (CERN experiment)

- Muon neutrinos were sent across a distance of 730 km
- At the speed of light, travelling that distance takes about 2.4 ms
- The neutrinos were observed to arrive 60 ns earlier

Formalisation:

1. Let c be the speed of light
2. In the experiment the distance is 730 km: dist(730km)
3. Let n be a neutrino; neutrinos are objects: obj(n)
4. Calculation: 730km/c = 2.4ms
5. Observation: travelTime(n,730km) = 2.4ms−60ns
6. Math: 2.4ms ≠ 2.4ms−60ns
7. Einstein: for all x, d. obj(x) implies (dist(d) implies travelTime(x,d) ≥ d/c)

Faster than light travel (CERN experiment)

1. Let c be the speed of light
2. In the experiment the distance is 730 km: dist(730km)
3. Let n be a neutrino; neutrinos are objects: obj(n)
4. Calculation: 730km/c = 2.4ms
5. Observation: travelTime(n,730km) = 2.4ms−60ns
6. Math: 2.4ms−60ns $\not\geq$ 2.4ms
7. Einstein:
   for all x, d. obj(x) implies (dist(d) implies travelTime(x,d) $\geq$ d/c)
8. By Instantiation:
   obj(n) implies (dist(730km) implies travelTime(n,730km) $\geq$ 730km/c)
9. By Modus Ponens:
   (dist(730km) implies travelTime(n,730km) $\geq$ 730km/c)
10. By Modus Ponens:
    travelTime(n,730km) $\geq$ 730km/c
11. By Equality:
    travelTime(n,730km) $\geq$ 2.4ms

Faster than light travel (CERN experiment)

1. Let c be the speed of light
2. In the experiment the distance is 730 km: dist(730km)
3. Let n be a neutrino; neutrinos are objects: obj(n)
4. Calculation: 730km/c = 2.4ms
5. Observation: travelTime(n,730km) = 2.4ms−60ns
6. Math: 2.4ms−60ns $\not\geq$ 2.4ms
7. Einstein:
   for all x, d. obj(x) implies (dist(d) implies travelTime(x,d) $\geq$ d/c)

   ⋮
11. By Equality:
    travelTime(n,730km) $\geq$ 2.4ms
12. By Equality:
    2.4ms−60ns $\geq$ 2.4ms

6 and 12 contradict each other $\rightsquigarrow$ one of 1 to 7 must be wrong
Closer analysis $\rightsquigarrow$ one of 2, 5, or 7 must be wrong

Newton versus Einstein

Newton's theory of gravity versus Einstein's theory of relativity

- Largely make the same predictions
- Both predict that the sun's gravity should bend rays of light
- However, Einstein's theory predicts a greater deflection
- Correctness of Einstein's prediction confirmed by observation in 1919

1. Observation: beam(b)
2. Observation: passesSun(b)
3. Observation: deflection(b) = d_e
4. Math: $d_e \neq d_n$
5. Einstein:
   for all x. beam(x) and passesSun(x) implies deflection(x) = d_e
6. Newton:
   for all x. beam(x) and passesSun(x) implies deflection(x) = d_n

Newton versus Einstein

1. Observation: beam(b)
2. Observation: passesSun(b)
3. Observation: deflection(b) = d_e
4. Math: $d_e \neq d_n$
5. Einstein:
   for all x. beam(x) implies (passesSun(x) implies deflection(x) = d_e)
6. Newton:
   for all x. beam(x) implies (passesSun(x) implies deflection(x) = d_n)
7. By Instantiation:
   beam(b) implies (passesSun(b) implies deflection(b) = d_n)
8. By Modus Ponens:
   passesSun(b) implies deflection(b) = d_n
9. By Modus Ponens:
   deflection(b) = d_n
10. By Equality:
    $d_e = d_n$

Newton versus Einstein

1. Observation: beam(b)
2. Observation: passesSun(b)
3. Observation: deflection(b) = d_e
4. Math: $d_e \neq d_n$
5. Einstein:
   for all x. beam(x) implies (passesSun(x) implies deflection(x) = d_e)
6. Newton:
   for all x. beam(x) implies (passesSun(x) implies deflection(x) = d_n)

   ⋮
10. By Equality:
    $d_e = d_n$

- 4 and 10 contradict each other $\rightsquigarrow$ one of 1 to 4 or 6 must be wrong
- Closer analysis $\rightsquigarrow$ 3 or 6 must be wrong
- Assuming we trust 3 and seeing that Einstein (5) would correctly predict 3, we conclude that 6 is wrong and 5 is right

Automated Deduction

Can deductive reasoning be (efficiently) automated?

- Automated Theorem Proving and Interactive Theorem Proving are the areas of Artificial Intelligence that provide answers to this and related questions
- Focus is on formal logic, for example,

  propositional logic, first-order logic, higher-order logic

  Calculi (sets of inference rules) include

  DPLL, tableaux calculi, sequent calculi, resolution calculi

  These are related to but not identical to the inference rules we have seen

Automated Deduction

Can deductive reasoning be (efficiently) automated?

- An automated theorem prover attempts to constructs a proof for a given hypothesis from a given set of premises with little / no user interaction

  Example systems:
  - E Schulz (Munich)
  - EQP McCune (Argonne)
  - SPASS Weidenbach et al (Saarbrücken)
  - Vampire Voronkov, Riazanov, Hoder (Manchester)
  - Waldmeister Hillenbrand (Saarbrücken) et al

  Example applications:
  - Robbins conjecture (open for 63 years) stating that a particular group of axioms forms a basis for Boolean algebra was solved by EQP
  - Spec# static verifier
    Spec# is a formal language for API contracts that extends Microsoft's C#
  - Waldmeister technology is used in Mathematica to simplify equations

Automated Deduction

Can deductive reasoning be (efficiently) automated?

- An interactive theorem prover or proof assistant assists a human user in the development of formal proofs;
  the human user guides the proof assistant, for example, by instructing the system which rule to apply to which premises

  Example systems:
  - Isabelle/HOL Paulson (Cambridge), Nipkow (München), Wenzel (Paris-Sud)
  - Twelf Pfenning (CMU) and Schürmann (ITU Copenhagen)

  Example application:
  - Verification of seL4, a microkernel consisting of 8,700 lines of C code and 600 lines of assembler
  - Machine-checking a large sublanguage of sequential Java, covering the type system and operational (evaluation) semantics of the language

Automation vs Interaction: Pros and Cons

Pros:

- Automated theorem provers can be turned into push button technology
  - User does not need to know how it works
  - Premises are automatically constructed by the system
  - Hypothesis is also either automatically constructed by the system or specified by the user in a language that the user finds understandable

Cons:

- Logics supported by automated theorem provers are typically not as expressive as those supported by interactive theorem provers
- Theorem proving tasks typically have a high computational complexity
  $\leadsto$ automated theorem provers may not always complete a task in a time acceptable to the user (or not at all)

Automation vs Interaction: Pros and Cons

Pros:

- Interactive theorem proving gives you (almost) the full power of mathematical proof without human fallacy
  - If something can be proved by a human mathematician then it should be provable using interactive theorem proving
  - Automated theorem provers can be integrated into interactive theorem provers
  - Logics supported by interactive theorem provers are typically more expressive than those supported by automated theorem provers

Cons:

- An average software developer is not able to use interactive theorem proving
- Interactive theorem proving can be a tedious long drawn-out process for humans

Intellectual Discovery: Deduction

- Deductive reasoning is often said not to lead to new knowledge
  (Note: This implies pure mathematicians largely waste their time)

  $\leadsto$ Seriously underestimates the computational effort involved in deductive reasoning

  $\leadsto$ Most theories are undecidable
  (There is no algorithm that even given infinite time could determine whether a statements follows from a theory or not)

  $\leadsto$ Thus, establishing that a statement follows from a theory extends our knowledge

Deduction: Definitions

Deductive reasoning (Logic)

- Deductive reasoning is a logical process that draws a conclusion from a set of premises in such a way that the conclusion is true whenever all premises are true
- Deductive reasoning uses a set of well-defined deductive rules (inference rules) for this logical process
- A "trace" of this logical process is a proof

Deductive reasoning (Argumentation)

- Deductive reasoning is reasoning which constructs or evaluates deductive arguments
- Deductive arguments are attempts to show that a conclusion necessarily follows from a set of premises

Proof versus Argument: Example

- Proof (sketch):
  - Jim was driving a car
  - The car was driving at 90 mph (miles per hour) on the motorway
  - The speed limit on motorways is 70 mph
  - 90 mph is greater than 70 mph
  - If a car exceeds the speed limit,
    then the driver should get penalty points
  - [. . .] (Instantiation, Equality, Modus Ponens)
  - Therefore, Jim should get penalty points
- Argument:
  - Jim should get penalty points
    because his car was exceeding the speed limit

Proof versus Argument (1)

- Arguments leave some assumptions implicit,
  for example, '90 mph is greater than 70 mph' or 'Jim was driving'
  (implicit assumptions are also called presuppositions)
- Arguments may use open texture,
  for example, there is no need to specify the speed limit
- Arguments can contain uncertain information,
  for example, in the argument we do not state the speed of Jim's car

Proof versus Argument (2)

- There can be arguments for A and also for not A,
  for example, for 'Jim should get penalty points'
  and for 'Jim should not get penalty points notice'
- Argument still need to be valid,
  that is, be constructed using deduction rules
- But arguments, unlike proofs, are always defeasible:
  new information can be brought to light that forces us to abandon the conclusion of an argument
- Therefore arguments can (always) be challenged

Challenges: Examples

Jim should get penalty points
because his car was exceeding the speed limit

- Jim should not get penalty points
  because he was not driving the car
  - Argument for the negation of the conclusion (defeater)
- Jim's car did not exceed the speed limit
  - Denies a premise
- It is wrong to punish driver's for speeding
  - Attacks the rule/theory:
    If a car exceeds the speed limit,
    then the driver should get penalty points
- The car is an ambulance and
  Jim was getting an injured person to a hospital
  - Argues that the rule is not applicable (undercutter);
    does so be stating that there should be exception to the rule

Challenges: Examples

Jim should get penalty points
because his car was exceeding the speed limit

- Jim should not get penalty points
  because he was not driving the car

Challenges are themselves arguments and can be challenged:

- Even if Jim was not driving he should still get penalty points
  because he is owner the of the car

But it is also possible to provide additional support for a particular conclusion:

- Jim should still get penalty points
  because he is a very bad driver
  and the sooner he loses his driving licence the better

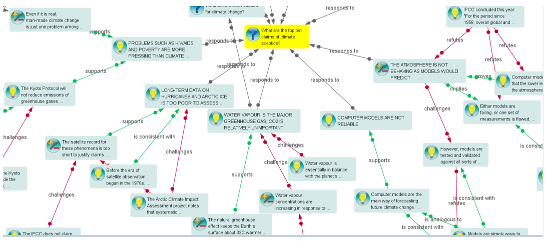
Argument status

- In considering an issue a proponent puts forward one or more arguments supplying a rational justification for a conclusion
- These arguments can be challenged with other arguments
- In turn, the new arguments can be challenged
- The result is a set of arguments and a set of attack relations between them
- We can then consider the following questions:
  - Which arguments must be accepted
  - Which arguments can be accepted
  - Which sets of arguments are acceptable together
  - Which arguments are indefensible

All this can be formalised in argumentation frameworks, and then analysed and implemented

Argumentation systems

- Atkinson's and Cartwright's 'Parmenides' system
  http://cgi.csc.liv.ac.uk/~parmenides/index.php
- Hoffmann et al's 'AGORA-net' system
  http://agora.gatech.edu/?page_id=257
- Shum's 'Cohere' system
  http://cohere.open.ac.uk/



Top ten arguments of Climate Change Sceptics and counter-arguments