Logical Argument Mapping (LAM) – A Manual

Michael Hoffmann, m.hoffmann@gatech.edu, January 2011

Please find the most recent version of this manual (including a list of publications) at http://lam.spp.gatech.edu. See also http://agora.gatech.edu/

Introduction

In order to understand someone's position or thesis, it is good to know this person's justification for this thesis or position. This is especially important for ethical problems. Since ethical decisions can be based on a variety of ethical principles and moral considerations, there are often good arguments for conflicting positions.

Justifications can be represented in the form of arguments. For example, if I want to argue for the thesis "Paul is responsible for what he did," I might provide as a justification for this thesis the reason "Paul is a rational being."

What is an argument? An argument is defined as a set of statements—a claim and one or more reasons—where the reasons jointly provide support (not necessarily conclusive) for the claim, or are at least intended to support the claim. In my example, the statement "Paul is responsible for what he did" would be the claim, and the justification "Paul is a rational being" is the reason.

Logical Argument Mapping (LAM) and the corresponding web-based software system "AGORA: Participate – Deliberate" are built on the idea that the visualization of arguments in graphical form facilitates the structuring of complex justifications and debates, and stimulates self-reflection. The reason for the former is that argument mapping helps us to represent entire argumentations, that is chains of arguments, including objections, counter-, and counter-counter-arguments. Based on the graphical structure of argument maps, the central claim, the structure of justifications, controversial points, open ends, and the status of complex debates are immediately visible. On an argument map, everything is clearly located at a certain position. Everything is part of a structure.

The reason for the second assumption that argument mapping stimulates self-reflection is the fact that we have to create this structure. We have to reflect on the adequacy of a certain structure, and we have to revise it if necessary. This helps us to reflect on our own thinking about an issue —and on that of others when we are using LAM to represent or reconstruct given arguments.

What distinguishes LAM and the AGORA approach from **other argument visualization tools** is the fact that it guides the user to represent arguments in the form of deductively valid arguments.

What is a deductively valid argument? An argument is "deductively valid" if and only if it follows an argument scheme that is deductively valid. An argument scheme is deductively valid if and only if it is impossible for any argument following this scheme to have true premises and a false conclusion. See, for example, the deductively valid argument scheme that is called *modus ponens*:

- *p*
- if *p*, then *q*
- therefore, *q*

Every argument that is formulated according to this scheme will be deductively valid (as long as *p* and *q* are variables that represent propositions, and "if *p*, then *q*" is understood as material implication, that is as something like a law of nature that connects an event described by *q* as a necessary consequence of an event described by *p)*. For example:

- Paul is a rational being
- if Paul is a rational being, then Paul is responsible for what he did
- therefore, Paul is responsible for what he did

This example shows that it is possible to transform any argument into a deductively valid argument simply by introducing a fitting additional premise like the “if-then” statement in this example. I call this additional premise an “enabler.” The “enabler” in an argument is the premise that guarantees that the reason provided is sufficient to justify the claim. The enabler “enables” the reason to produce the claim with logical necessity. Thus, the simple argument “Paul is a rational being, therefore he is responsible for what he did” can be transformed into a deductively valid argument by constructing the enabler “if Paul is a rational being, then Paul is responsible for what he did.”

In contrast to classical deduction, in LAM deductively valid arguments are interpreted as *defeasible* deductions. Even though -- as in classical deductive validity -- a conclusion will be necessarily true in case the premises are true, in LAM both the enabler and the reason of an argument are only *believed* to be true by the person proposing the argument, and only as long as there is no information to the contrary. If information comes up that would either defeat or question one of the premises, this information will be connected to this premise as an "objection," and the status of every proposition that depends on this defeated or questioned premise will change from "undefeated" to "defeated" or "questioned" (see p.7 for an example). This way, an entire deductive argument can be defeated by defeating one of its premises.

LAM and the AGORA system use seven deductively valid argument schemes: *modus ponens*; *modus tollens*; disjunctive syllogism; not-both syllogism; conditional syllogism; equivalence; and constructive dilemma (see Section 3 for details). This list is the result of a compromise between completeness and practicality. There are more deductively valid argument schemes, but these turned out to be the ones whose validity is easily comprehensible, and that are sufficient to represent most of the arguments that we are using every day (after a fitting enabler has been introduced).

Why should it make sense to transform arguments into defeasible deductions? There are three reasons for this fundamental design decision:

1. A thesis about human cognition: Critical reflection and learning can be better achieved with those systems of representation that provide a clear normative standard of argument construction that constrains the freedom of expression; a standard that challenges the user to be more specific than he would be otherwise, to slow down and think more thoroughly.
2. A consideration from argumentation theory: In order to locate any possible objection against an argument precisely, anything that can be criticized in an argument must be represented. The easiest way to achieve this form of completeness is to present an argument in deductively valid form. Looking at a deductively complete argument reminds us that we do not only have to reflect on the question whether the reasons we provide are acceptable and justified, but also the inferential relation between reason and claim.
3. An educational and computational argument: Learning needs scaffolding, and software tools that

are designed to support autonomous learning--either individually or in groups--should guide the user in a step-by-step process. This can be much easier achieved by software tools whose means of expression are limited to deductive argument schemes.

Logical Argument Mapping is not deductive reasoning. Logical Argument Mapping is the process of *constructing* arguments in deductive form, *assessing* the acceptability of the premises as they need to be formulated to achieve this deductive form, and *revising* these premises and/or the structure of the argument as long as it takes to construct the best possible argument. A reconstruction of an argument in logical form can show us how its premises would need to look like if the goal were to guarantee the truth of the conclusion. The point is to get the content of the premises right and to formulate them in their strongest possible form. For an example of how such a process of improving an argument might work, see the “Tweety can fly” example below, on the page about “Defeasible reasoning in a dialogue.”

1. Three basic rules

1. Represent your main argument—and every sub-argument that might be controversial—according to an argument scheme whose deductive validity is evident or can be made plausible (see section 3 for a list)
2. Consider the acceptability of all your premises, and provide further arguments for those whose acceptability is either not evident or controversial
3. Make sure that all your premises are consistent with each other.

2. The procedure of Logical Argument Mapping

You can either start with constructing your own argument (or reconstructing a given argument) or with a distinction of different cases as described in Section 2.3 below. In the latter case, you would first produce a list of claims for which arguments have to be developed.

Logical Argument Mapping and the AGORA system allow the seven kinds of activities that are listed below. The first two—argument construction and evaluation—are necessary elements, the remaining five are optional. Every LAM procedure must include the construction (or re-construction) of an argument and its evaluation. The construction is constrained by a set of rules (Sect. 1 above), and the evaluation needs to make sure that these three rules are fulfilled.

1. Argument construction

The following sequence of steps assumes that the goal is to construct an argument. For the reconstruction of a given argument it is important, first of all, to identify the central claim. It should always be possible to describe the central argument in a few sentences. If your reconstruction of the central arguments gets too complex, you might be on the wrong track with your interpretation.

1. Formulate a claim: the central goal of your argument, a central thesis. Decide whether your claim is a universal statement (“cheating is wrong”) or a particular statement (“in case X, cheating is justified”). See the LAM conventions below for how to represent these possibilities.
2. Provide a reason for your claim, or a combination of reasons that together are sufficient to justify your claim (i.e., a simple or linked argument).

3. Select from a list of argument schemes whose logical validity you accept the scheme that is most adequate for your argument (see Section 3).
4. Transform your argument into a logical argument by adding what is missing, and by reformulating the elements of the argument (claim, reason, enabler) in such a way that its validity in accordance with the scheme becomes evident.
5. Consider possible objections against both the reason(s) and the enabler, formulate them, and link them to the elements of your map against which they are directed (see section 3 for some "objection schemes" that you can use for this purpose).
6. Decide whether to
   a) develop new arguments against the objections, or
   b) reformulate the original argument in such a way that it can be defended against the objection by, e.g.,
      - including exceptions into the enabler and limiting the scope of the claim (go back to step 1. or 2.), or
      - using a different argument scheme (go to step 3.), or
      - redefining the meaning of concepts used in the argument (go to step 1. or 2.), or
   c) give up the entire argument
7. In case of 6.c, start again with step 1. or 2.; in the other cases, do as described in 6.a and b.
8. Consider further reasons for your claim and perform steps 3. to 7. for them as well.

2. Argument evaluation

The following criteria allow the evaluation of argument maps. Evaluation is important since it is always possible to represent a text or an issue in many different ways. Evaluation should motivate the revision or refinement of an argument map.

1. **Validity**: Central and controversial arguments must be formulated in deductively valid form, that is in correspondence to the argument schemes listed in Section 3.
2. **Acceptability**: Check each claim in your text boxes and ask yourself whether you can accept it as it is formulated. If the claim is too complex, or if it is hard to see whether it is acceptable or not, reformulate or divide into separate claims. This is especially important when you are reconstructing someone else's argumentation and you assume claims that you cannot directly quote from your source. It is easy to write something down, but you will never be able to defend it if it is either nonsense or hardly acceptable. If a claim is not acceptable, revise the entire argument; if it should be acceptable based on further arguments, then develop these arguments to support it.
3. **Simplicity**: Generally, the simpler an argumentation the more convincing. The criterion of simplicity should motivate you to focus from the very beginning on the essential message of your argumentation. Don't get confused by too much detail and things that are only marginally important. Work from the center to the margins, and do so only when you are convinced that you found the best possible form for the center of your argumentation. Then focus on supporting the reasons of your central argument and on defending these reasons

against possible objections.

4. **Balance**: The stronger a position, the weaker is often the argument for it, and the weaker a position, the easier it is to formulate a strong argument. Finding here the right balance is crucial. Everything depends on how you phrase the final conclusion of your argumentation. Experiment with different formulations and try to develop arguments that are strong enough for your position.

3. Classification of possibilities or options

Sometimes it is necessary to distinguish different cases for a certain claim so that arguments or objections can be developed for each case. This can be done as described in the "LAM conventions" (Section 3) under "topographical structure of typologies and classifications."

4. Objections

Different forms of objections to specified elements of an argumentation can be represented by a variety of "objection schemes" (ObjScheme). Their main function is to motivate the improvement or revision of an argumentation (see sect. 3 for a list). Objections can only be directed against the premises of an argument (reasons and enabler). It is of course possible to develop an independent argument against the conclusion. However, since this conclusion is necessarily true if all the premises are true, it is always necessary to attack these premises if you don't agree with the conclusion. If you argue only against the conclusion of an argument, you have to create a new argument.

5. Questions, comments, definitions, examples, etc.

Sometimes it is useful to add questions, comments, definitions, examples, and other statements to the elements of an argument. These can be added by means of the non-logical connectors that are listed in the LAM conventions.

6. Supporting data

It is possible to add further information and supporting data in LAM maps.

7. Argument revision

In order to represent the development of an argumentation, it might be necessary to show how certain arguments or objections lead to revisions of parts of an argumentation. For this, LAM offers a set of "revision schemes" (see http://tinyurl.com/233zdua for some preliminary considerations).

The list of "revisions" that can be represented in LAM specifies different possibilities of revising either individual statements or the structure of arguments. Since the specification of revisions is something that we do with regard to an already given argumentation, representing revisions in a map means that both an argument and a meta-level of reflecting on the argumentation are represented in the same map.

3. Schemes

Please find examples of LAM maps at http://lam.spp.gatech.edu. In its current version, all LAM maps are created with Cmap, http://cmap.ihmc.us/.

Logical Argument Mapping (LAM)

see → http://lam.spp.gatech.edu

Michael Hoffmann
Georgia Institute of Technology
last modified: Aug 27, 2010

LAM conventions

form of textboxes

particular statement

universal statement

"Universal statement" is defined as a statement that can be falsified by one counterexample. Thus, laws, rules, and all statements that include "ought," "should," or other forms indicating normativity, are universal statements. Anything else is treated as a "particular statement," including statements about possibilities. The distinction is important only with regard to the consequences of different forms of objections: If a premise is *defeated,* then the conclusion and every chain of arguments that depends on this premise is defeated as well; but if a premise is only *questioned* or criticized, then the conclusion and everything depending is only questioned, but not defeated. While universal statements can easily be defeated by one counterexample, it depends on an agreement among deliberators whether a counterargument against a particular statement is sufficient to defeat it, even though it is always sufficient to question it and to shift, thus, the burden of proof.

topographical structure of logically valid arguments

a simple argument

claim ← therefore *(ArgScheme: X AU: Y)* — reason
enabler

two independent arguments for the same claim

claim ← therefore *(ArgScheme: X AU: Y)* — reason 1
enabler 1

claim ← therefore *(ArgScheme: X AU: Y)* — reason 2
enabler 2

a linked argument: both reasons and an enabler that refers to both are necessary to guarantee the truth of the claim

claim ← therefore *(ArgScheme: X AU: Y)* — reason 1, reason 2
enabler

an "enabler" is the premise in an argument that guarantees that the reason provided is sufficient to justify the claim. The enabler is always a universal statement. Enablers are crucial because they are assumed to represent parts of an arguer's implicit background assumptions.

topographical structure of typologies and classifications

claim — includes → 1. claim, 2. claim, 3. claim

concept/ type — includes → 1. sub-type, 2. sub-type, 3. sub-type

color of textboxes

a consistent argumentation

another consistent argumentation, or parts of it

comments on the mapping process

other colors for further positions, or parts of them

layout convention

the central claim of an argumentation is located at the top left corner of a map

non-logical connectors between text boxes

"supports"; "leads to"; "defined as"; "depends on"; "based on"; "for example"; "leads to the question"; "answers"; "comments on"; "friendly amendment" (to suggest a reformulation of an author's claim); ...

the small arrows at the bottom of text boxes are links to arguments

the small icons at the bottom of text boxes are links to resources

"ArgScheme": "argument scheme." Refers here to a one of the logical valid schemes whose conclusion is necessarily true if all the premises are true

"ObjScheme": objection scheme." Refers to one of the schemes by which objections and defeat relations can be expressed in LAM

"AU": "author" (distinguishing authors is important for representing controversies)

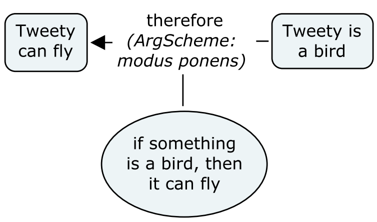
Tweety

Defeasible reasoning in a dialogue

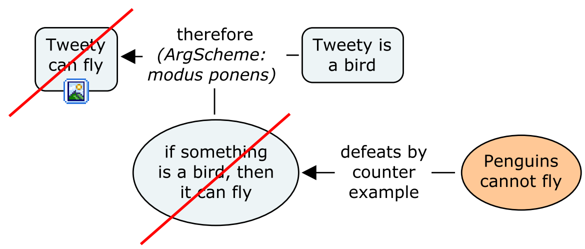
Proponent

Opponent

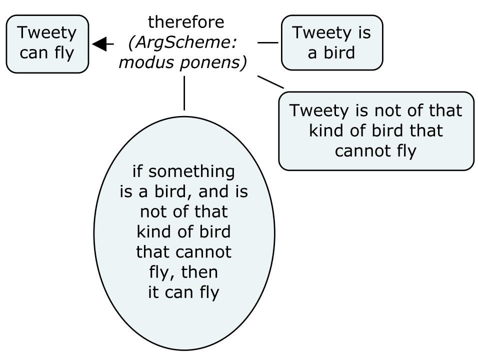
1.



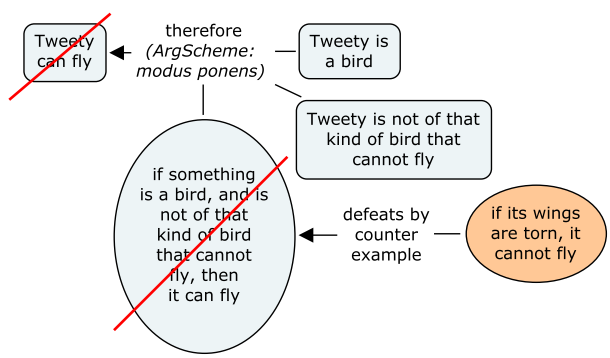
2.



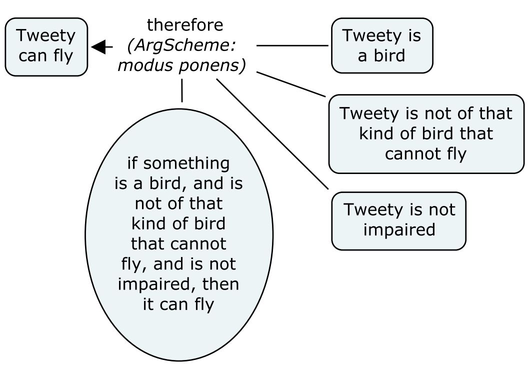
3.



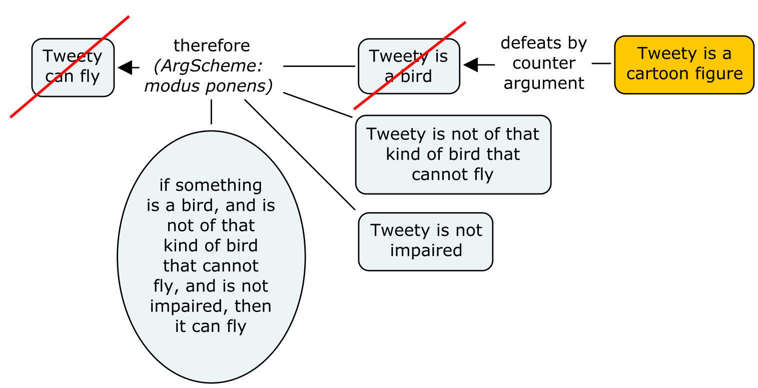
4.

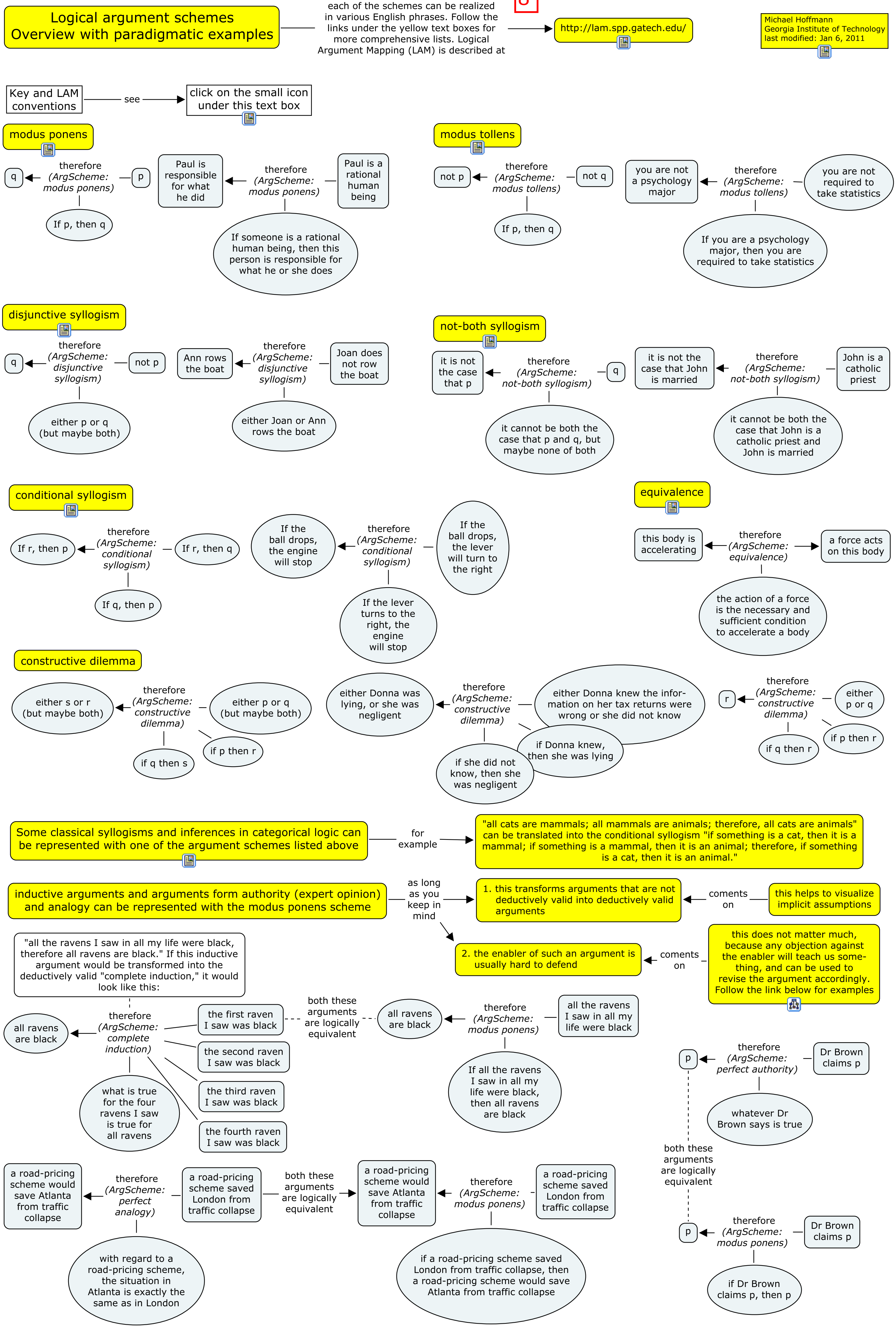


5.

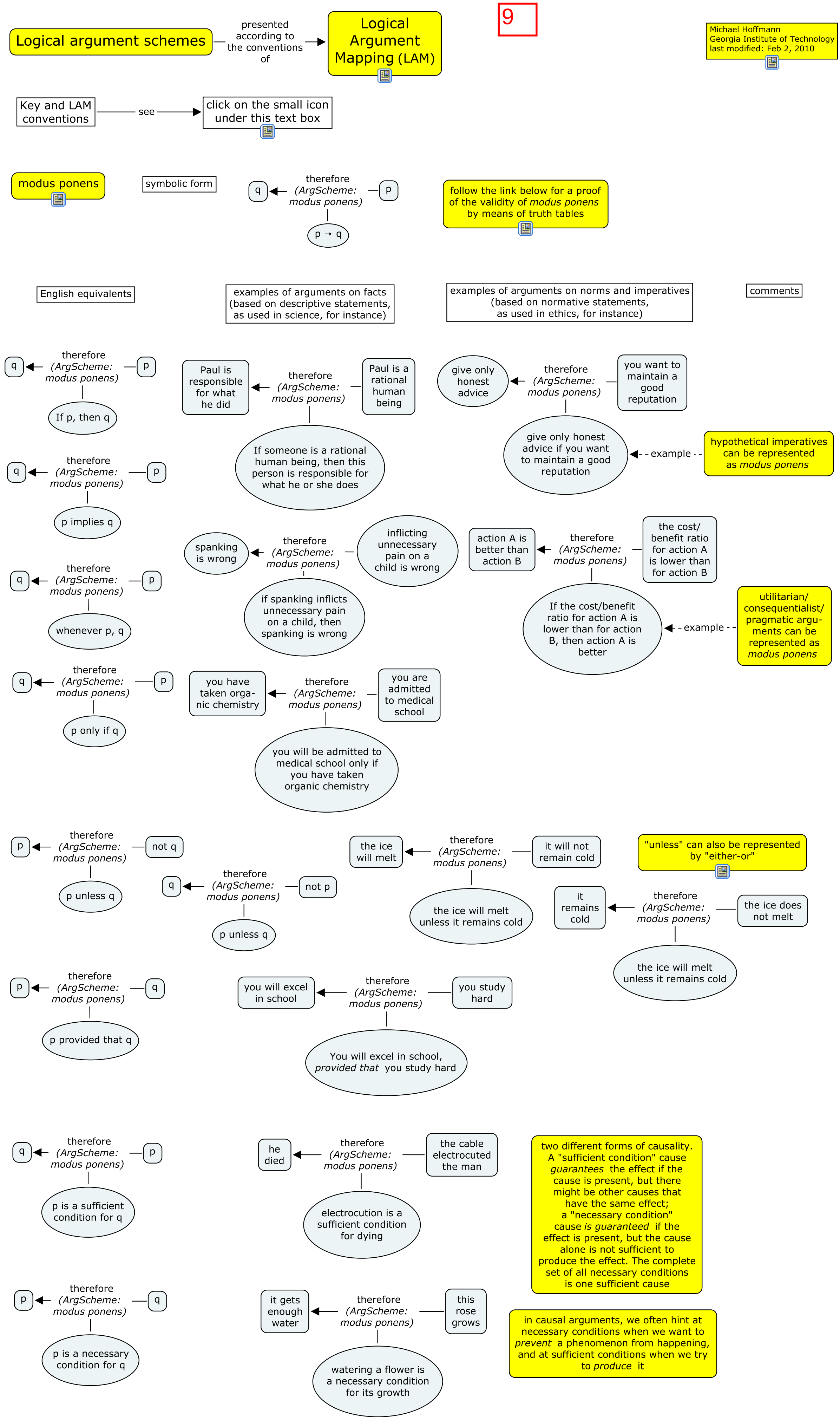


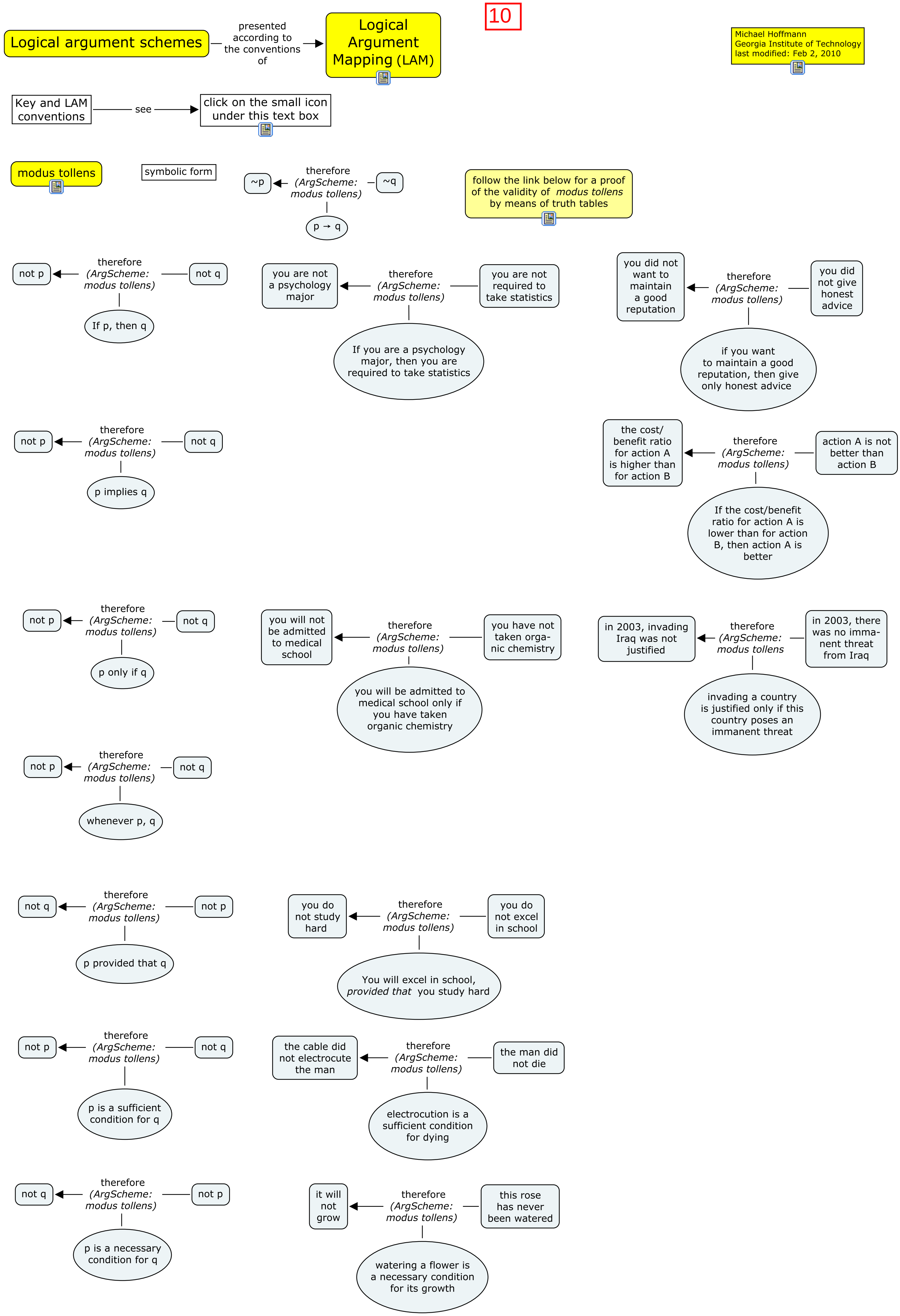
6.





Logical argument schemes
Overview with paradigmatic examples
each of the schemes can be realized in various English phrases. Follow the links under the yellow text boxes for more comprehensive lists. Logical Argument Mapping (LAM) is described at
http://lam.spp.gatech.edu/
Michael Hoffmann
Georgia Institute of Technology
last modified: Jan 6, 2011
Key and LAM conventions
see
click on the small icon under this text box
modus ponens
q
therefore (ArgScheme: modus ponens)
p
If p, then q
Paul is responsible for what he did
therefore (ArgScheme: modus ponens)
Paul is a rational human being
If someone is a rational human being, then this person is responsible for what he or she does
modus tollens
not p
therefore (ArgScheme: modus tollens)
not q
If p, then q
you are not a psychology major
therefore (ArgScheme: modus tollens)
you are not required to take statistics
If you are a psychology major, then you are required to take statistics
disjunctive syllogism
q
therefore (ArgScheme: disjunctive syllogism)
not p
either p or q (but maybe both)
Ann rows the boat
therefore (ArgScheme: disjunctive syllogism)
Joan does not row the boat
either Joan or Ann rows the boat
not-both syllogism
it is not the case that p
therefore (ArgScheme: not-both syllogism)
q
it cannot be both the case that p and q, but maybe none of both
it is not the case that John is married
therefore (ArgScheme: not-both syllogism)
John is a catholic priest
it cannot be both the case that John is a catholic priest and John is married
conditional syllogism
If r, then p
therefore (ArgScheme: conditional syllogism)
If r, then q
If q, then p
If the ball drops, the engine will stop
therefore (ArgScheme: conditional syllogism)
If the ball drops, the lever will turn to the right
If the lever turns to the right, the engine will stop
equivalence
this body is accelerating
therefore (ArgScheme: equivalence)
a force acts on this body
the action of a force is the necessary and sufficient condition to accelerate a body
constructive dilemma
either s or r (but maybe both)
therefore (ArgScheme: constructive dilemma)
either p or q (but maybe both)
if q then s
if p then r
either Donna was lying, or she was negligent
therefore (ArgScheme: constructive dilemma)
either Donna knew the information on her tax returns were wrong or she did not know
if she did not know, then she was negligent
if Donna knew, then she was lying
r
therefore (ArgScheme: constructive dilemma)
either p or q
if q then r
if p then r
Some classical syllogisms and inferences in categorical logic can be represented with one of the argument schemes listed above
for example
"all cats are mammals; all mammals are animals; therefore, all cats are animals" can be translated into the conditional syllogism "if something is a cat, then it is a mammal; if something is a mammal, then it is an animal; therefore, if something is a cat, then it is an animal."
inductive arguments and arguments form authority (expert opinion) and analogy can be represented with the modus ponens scheme
as long as you keep in mind
1. this transforms arguments that are not deductively valid into deductively valid arguments
coments on
this helps to visualize implicit assumptions
2. the enabler of such an argument is usually hard to defend
coments on
this does not matter much, because any objection against the enabler will teach us something, and can be used to revise the argument accordingly. Follow the link below for examples
"all the ravens I saw in all my life were black, therefore all ravens are black." If this inductive argument would be transformed into the deductively valid "complete induction," it would look like this:
all ravens are black
therefore (ArgScheme: complete induction)
the first raven I saw was black
the second raven I saw was black
the third raven I saw was black
the fourth raven I saw was black
what is true for the four ravens I saw is true for all ravens
both these arguments are logically equivalent
all ravens are black
therefore (ArgScheme: modus ponens)
all the ravens I saw in all my life were black
If all the ravens I saw in all my life were black, then all ravens are black
p
therefore (ArgScheme: perfect authority)
Dr Brown claims p
whatever Dr Brown says is true
both these arguments are logically equivalent
p
therefore (ArgScheme: modus ponens)
Dr Brown claims p
if Dr Brown claims p, then p
a road-pricing scheme would save Atlanta from traffic collapse
therefore (ArgScheme: perfect analogy)
a road-pricing scheme saved London from traffic collapse
with regard to a road-pricing scheme, the situation in Atlanta is exactly the same as in London
both these arguments are logically equivalent
a road-pricing scheme would save Atlanta from traffic collapse
therefore (ArgScheme: modus ponens)
a road-pricing scheme saved London from traffic collapse
if a road-pricing scheme saved London from traffic collapse, then a road-pricing scheme would save Atlanta from traffic collapse


Logical argument schemes
presented according to the conventions of
Logical Argument Mapping (LAM)
Michael Hoffmann
Georgia Institute of Technology
last modified: Feb 2, 2010
Key and LAM conventions
see
click on the small icon under this text box
modus ponens
symbolic form
q
therefore (ArgScheme: modus ponens)
p
p → q
follow the link below for a proof of the validity of modus ponens by means of truth tables
English equivalents
examples of arguments on facts (based on descriptive statements, as used in science, for instance)
examples of arguments on norms and imperatives (based on normative statements, as used in ethics, for instance)
comments
q
therefore (ArgScheme: modus ponens)
p
If p, then q
Paul is responsible for what he did
therefore (ArgScheme: modus ponens)
Paul is a rational human being
If someone is a rational human being, then this person is responsible for what he or she does
give only honest advice
therefore (ArgScheme: modus ponens)
you want to maintain a good reputation
give only honest advice if you want to maintain a good reputation
example
hypothetical imperatives can be represented as modus ponens
q
therefore (ArgScheme: modus ponens)
p
p implies q
spanking is wrong
therefore (ArgScheme: modus ponens)
inflicting unnecessary pain on a child is wrong
if spanking inflicts unnecessary pain on a child, then spanking is wrong
action A is better than action B
therefore (ArgScheme: modus ponens)
the cost/ benefit ratio for action A is lower than for action B
If the cost/benefit ratio for action A is lower than for action B, then action A is better
example
utilitarian/ consequentialist/ pragmatic argu-ments can be represented as modus ponens
q
therefore (ArgScheme: modus ponens)
p
whenever p, q
q
therefore (ArgScheme: modus ponens)
p
p only if q
you have taken orga-nic chemistry
therefore (ArgScheme: modus ponens)
you are admitted to medical school
you will be admitted to medical school only if you have taken organic chemistry
p
therefore (ArgScheme: modus ponens)
not q
p unless q
q
therefore (ArgScheme: modus ponens)
not p
p unless q
the ice will melt
therefore (ArgScheme: modus ponens)
it will not remain cold
the ice will melt unless it remains cold
"unless" can also be represented by "either-or"
it remains cold
therefore (ArgScheme: modus ponens)
the ice does not melt
the ice will melt unless it remains cold
p
therefore (ArgScheme: modus ponens)
q
p provided that q
you will excel in school
therefore (ArgScheme: modus ponens)
you study hard
You will excel in school, provided that you study hard
q
therefore (ArgScheme: modus ponens)
p
p is a sufficient condition for q
he died
therefore (ArgScheme: modus ponens)
the cable electrocuted the man
electrocution is a sufficient condition for dying
two different forms of causality. A "sufficient condition" cause guarantees the effect if the cause is present, but there might be other causes that have the same effect; a "necessary condition" cause is guaranteed if the effect is present, but the cause alone is not sufficient to produce the effect. The complete set of all necessary conditions is one sufficient cause
p
therefore (ArgScheme: modus ponens)
q
p is a necessary condition for q
it gets enough water
therefore (ArgScheme: modus ponens)
this rose grows
watering a flower is a necessary condition for its growth
in causal arguments, we often hint at necessary conditions when we want to prevent a phenomenon from happening, and at sufficient conditions when we try to produce it


Logical argument schemes
presented according to the conventions of
Logical Argument Mapping (LAM)
Michael Hoffmann
Georgia Institute of Technology
last modified: Feb 2, 2010
Key and LAM conventions
see
click on the small icon under this text box
modus tollens
symbolic form
~p
therefore (ArgScheme: modus tollens)
~q
p → q
follow the link below for a proof of the validity of modus tollens by means of truth tables
not p
therefore (ArgScheme: modus tollens)
not q
If p, then q
you are not a psychology major
therefore (ArgScheme: modus tollens)
you are not required to take statistics
If you are a psychology major, then you are required to take statistics
you did not want to maintain a good reputation
therefore (ArgScheme: modus tollens)
you did not give honest advice
if you want to maintain a good reputation, then give only honest advice
not p
therefore (ArgScheme: modus tollens)
not q
p implies q
the cost/ benefit ratio for action A is higher than for action B
therefore (ArgScheme: modus tollens)
action A is not better than action B
If the cost/benefit ratio for action A is lower than for action B, then action A is better
not p
therefore (ArgScheme: modus tollens)
not q
p only if q
you will not be admitted to medical school
therefore (ArgScheme: modus tollens)
you have not taken orga-nic chemistry
you will be admitted to medical school only if you have taken organic chemistry
in 2003, invading Iraq was not justified
therefore (ArgScheme: modus tollens
in 2003, there was no imma-nent threat from Iraq
invading a country is justified only if this country poses an immanent threat
not p
therefore (ArgScheme: modus tollens)
not q
whenever p, q
not q
therefore (ArgScheme: modus tollens)
not p
p provided that q
you do not study hard
therefore (ArgScheme: modus tollens)
you do not excel in school
You will excel in school, provided that you study hard
not p
therefore (ArgScheme: modus tollens)
not q
p is a sufficient condition for q
the cable did not electrocute the man
therefore (ArgScheme: modus tollens)
the man did not die
electrocution is a sufficient condition for dying
not q
therefore (ArgScheme: modus tollens)
not p
p is a necessary condition for q
it will not grow
therefore (ArgScheme: modus tollens)
this rose has never been watered
watering a flower is a necessary condition for its growth

Logical argument schemes — presented according to the conventions of → Logical Argument Mapping (LAM)

Michael Hoffmann
Georgia Institute of Technology
last modified: Dec 28, 2010

Key and LAM conventions — see → click on the small icon under this text box

symbolic form | English equivalents | examples of arguments on facts (based on descriptive statements, as used in science, for instance) | comments

disjunctive syllogism

to construct a disjunctive syllogism, start with the inference rule: "either p or q." In the disjunctive syllogism, the meaning of "either ... or" refers to an *inclusive* "or," that means: It is possible that both the p and the q are true. For example: When you say "I buy bread or milk," the whole sentence is still true when you buy both. If you use an *exclusive* "or," the sentence would be false if you buy both. This implies that you can use the disjunctive syllogism only when you know that either p or q is false. In this case, you can infer that the other is true. A disjunctive syllogism can include more than two elements in the enabler besides p and q, but you infer always a positive statement from negative ones

the validity of the disjunctive syllogism is based on the logical definition of "either-or."Click the link below to see the corresponding truth table definition

q ← therefore *(ArgScheme: disjunctive syllogism)* — ¬p; p v q

q ← therefore *(ArgScheme: disjunctive syllogism)* — not p; either p or q (but maybe both)

Ann rows the boat ← therefore *(ArgScheme: disjunctive syllogism)* — Joan does not row the boat; either Joan or Ann will row the boat

Ann rows the boat ← therefore *(ArgScheme: disjunctive syllogism)* — Joan does not row the boat; Joan rows unless Ann rows

p ← therefore *(ArgScheme: disjunctive syllogism)* — ¬q; p v q

p ← therefore *(ArgScheme: disjunctive syllogism)* — not q; either p or q (but maybe both)

Joan rows the boat ← therefore *(ArgScheme: disjunctive syllogism)* — Ann does not row the boat; either Joan or Ann rows the boat

Joan rows the boat ← therefore *(ArgScheme: disjunctive syllogism)* — Ann does not row the boat; Joan rows unless Ann rows

not-both syllogism

to construct a not-both syllogism, start with the inference rule. If you know that either p or q is true you can infer that the other proposition must be false. That means, there are two possibilities to formulate an argument, but both infer a negative statement from a positive one

the validity of the not-both syllogism is based on the logical definition of "not both." Click the link below to see the corresponding truth table definition

¬p ← therefore *(ArgScheme: not-both syllogism)* — q; p | q

it is not the case that p ← therefore *(ArgScheme: not-both syllogism)* — q; it cannot be both the case that p and q, but maybe none of both

it is not the case that John is married ← therefore *(ArgScheme: not-both syllogism)* — John is a catholic priest; it cannot be both the case that John is a catholic priest and John is married

¬q ← therefore *(ArgScheme: not-both syllogism)* — p; p | q

it is not the case that q ← therefore *(ArgScheme: not-both syllogism)* — p; it cannot be both the case that p and q, but maybe none of both

it is not the case that John is a catholic priest ← therefore *(ArgScheme: not-both syllogism)* — John is married; it cannot be both the case that John is a catholic priest and John is married

Logical argument schemes — presented according to the conventions of → Logical Argument Mapping (LAM)

Michael Hoffmann
Georgia Institute of Technology
last modified: Mar 25, 2010

Key and LAM conventions — see → click on the small icon under this text box

equivalence

to construct an argument based on equivalence, you would usually start with one of the four equivalent inference rules listed below. For each of them, you have four possibilities to formulate an argument:

q ← therefore *(ArgScheme: equivalence)* — p | p ↔ q

p ← therefore *(ArgScheme: equivalence)* — q | p ↔ q

not-q ← therefore *(ArgScheme: equivalence)* — not-p | p ↔ q

not-p ← therefore *(ArgScheme: equivalence)* — not-q | p ↔ q

English equivalents

examples of arguments on facts (based on descriptive statements, as used in science, for instance)

examples of arguments on norms and imperatives (based on normative statements, as used in ethics, for instance)

comments

q ← therefore *(ArgScheme: equivalence)* — p | p if and only if q

you clean up your room ← therefore *(ArgScheme: equivalence)* — you watch TV | you may watch TV if and only if you clean up your room

q ← therefore *(ArgScheme: equivalence)* — p | p just in case q

Mary is a teenager ← therefore *(ArgScheme: equivalence)* — she is 13 to 19 years of age | Mary is a teenager just in case she is 13 to 19 years of age

q ← therefore *(ArgScheme: equivalence)* — p | p is a necessary and sufficient condition for q

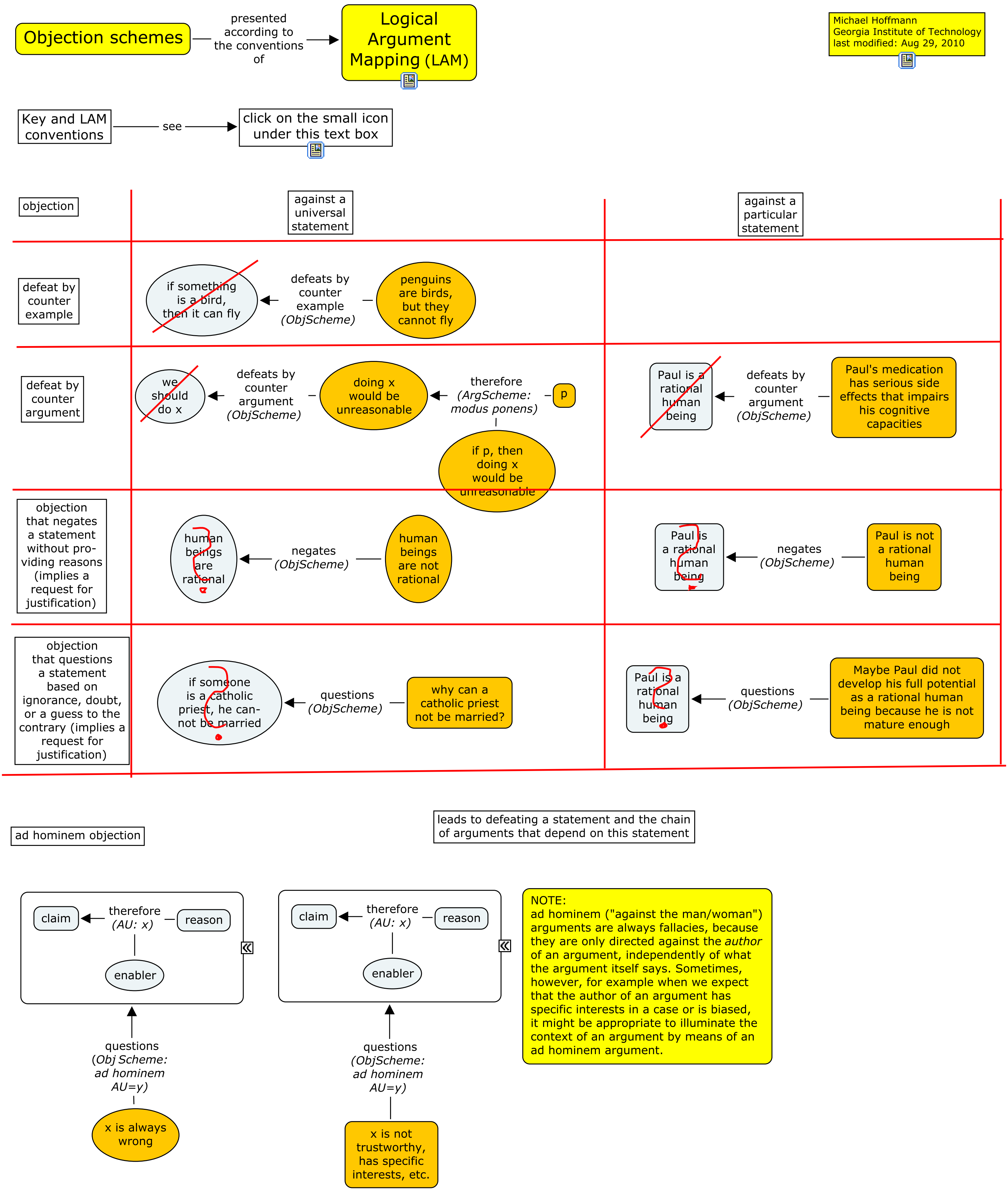
this body is accelerating ← therefore *(ArgScheme: equivalence)* — a force acts on this body | the action of a force is the necessary and sufficient condition to accelerate a body

the third form of causality (after what is described under *modus ponens)* . There is only one cause, i.e. a necessary condition which is at the same time sufficient to produce the effect

q ← therefore *(ArgScheme: equivalence)* — p | p and q are equivalent

2+2+2+2=8 ← therefore *(ArgScheme: equivalence)* — 4x2=8 | 2+2+2+2 and 4x2 are equivalent

you should say the truth ← therefore *(ArgScheme: equivalence)* — you should not lie | not lying and saying the truth are equivalent


Objection schemes
presented according to the conventions of
Logical Argument Mapping (LAM)
Michael Hoffmann
Georgia Institute of Technology
last modified: Aug 29, 2010
Key and LAM conventions
see
click on the small icon under this text box
objection
against a universal statement
against a particular statement
defeat by counter example
if something is a bird, then it can fly
defeats by counter example (ObjScheme)
penguins are birds, but they cannot fly
defeat by counter argument
we should do x
defeats by counter argument (ObjScheme)
doing x would be unreasonable
therefore (ArgScheme: modus ponens)
p
if p, then doing x would be unreasonable
Paul is a rational human being
defeats by counter argument (ObjScheme)
Paul's medication has serious side effects that impairs his cognitive capacities
objection that negates a statement without pro-viding reasons (implies a request for justification)
human beings are rational
negates (ObjScheme)
human beings are not rational
Paul is a rational human being
negates (ObjScheme)
Paul is not a rational human being
objection that questions a statement based on ignorance, doubt, or a guess to the contrary (implies a request for justification)
if someone is a catholic priest, he can-not be married
questions (ObjScheme)
why can a catholic priest not be married?
Paul is a rational human being
questions (ObjScheme)
Maybe Paul did not develop his full potential as a rational human being because he is not mature enough
ad hominem objection
leads to defeating a statement and the chain of arguments that depend on this statement
claim
therefore (AU: x)
reason
enabler
questions (Obj Scheme: ad hominem AU=y)
x is always wrong
claim
therefore (AU: x)
reason
enabler
questions (ObjScheme: ad hominem AU=y)
x is not trustworthy, has specific interests, etc.
NOTE:
ad hominem ("against the man/woman") arguments are always fallacies, because they are only directed against the author of an argument, independently of what the argument itself says. Sometimes, however, for example when we expect that the author of an argument has specific interests in a case or is biased, it might be appropriate to illuminate the context of an argument by means of an ad hominem argument.