ENTAILMENT AND PROPOSITIONAL LOGIC

Entailment in Arithmetic Formulas

But not vice versa

x + y < 6 **entail** x < 10

All models here → Is a model here as well

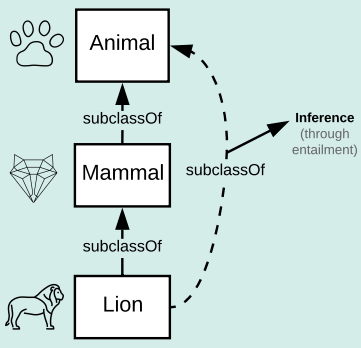
Model: An instance (i.e., value-assigned version) of a formula that is true.

Entailment in Concept Hierarchies

Concept: An abstraction or generalization arrived through experiences or transformation of existing ideas.

Lion subclassOf Mammals

Everything that is true for mammals must be true for all lions, but not everything true for lions is true for all mammals.



Universe:
{object1, object2, object3, ...}

Knowledge base:
{axiom1, axiom2, axiom3, ...}

Axiom:
Lion subclassOf Mammal

This statement becomes a *model* of the *axiom* only if the model is true.

Model:
lionNamedX subclassOf Mammals

Not a model (and not a counterexample):
snakeNamedY subclassOf Mammals

This statement is not true; therefore, it is not a model. If it was true, then it would have been a counterexample (but it is not).

Entailment in Propositional Logic

"True" sign / "Models" sign

Premises, Conditions, Knowledge base | Conclusion

$p \wedge \neg q \rightarrow r,\ \neg r,\ p \models q$

p: Train is late | q: There are taxis | r: Jane is late

Truth table is used to check whether this statement holds.

Not relevant for entailment

p	q	r	¬q	p∧¬q	p∧¬q→r	¬r
T	T	T	F	F	T	F
T	T	F	F	F	F	T
T	F	T	T	T	T	F
T	F	F	T	T	F	T
F	T	T	F	F	T	F
F	T	F	F	F	T	T
F	F	T	T	F	T	F
F	F	F	T	F	T	T

Entailment

Entailment condition: When the premises are true, the conclusion must be true (in all cases).

Each row is a ***valuation*** of ***variable***s 'p','q','r'

If there was a counterexample:

T	F				T	T

Entailment condition: When the premise is true, the conclusion must be true (in all cases).

Does p V q entail p → q ?

p	q	p v q	p->q	
T	T	T	T	✓ Entailment condition met
T	F	T	F	X Cond. not met (Counterexam.)
F	T	T	T	✓ Entailment condition met
F	F	F	T	Irrelevant, as premise is F

Counterexample

$\phi_1, \ldots, \phi_n \not\models \psi$

Truth Table

Not all lines in a truth table make sense. Establishing validity is to ascertain which ones do.

p	q	r	¬q	p∨¬q	p∨¬q→r
T	T	T	F	T	T
T	T	F	F	T	F
T	F	T	T	T	T
T	F	F	T	T	F
F	T	T	F	F	T
F	T	F	F	F	T
F	F	T	T	T	T
F	F	F	T	T	F

e.g., Train is late (p), and there are no taxis (-q), but in one case Jane is late (r) and in the other she is not late.

Equivalence

$p \rightarrow q \equiv \neg p \vee q$

p	q	p→q	¬p	¬p∨q
T	T	T	F	T
T	F	F	F	F
F	T	T	T	T
F	F	T	T	T

Same

Exclusive 'OR'

ϕ	ψ	ϕ ? ψ
T	T	F
T	F	T
F	T	T
F	F	F

Disjunction ('OR')

ϕ	ψ	ϕ ? ψ
T	T	T
T	F	T
F	T	T
F	F	F

Tautology

p	q	q→p	p→(q→p)
T	T	T	T
T	F	T	T
F	T	F	T
F	F	T	T

Implication

ϕ	ψ	ϕ ? ψ
T	T	T
T	F	F
F	T	T
F	F	T

*If you start out with a true premise, then the implication should be true only when the conclusion is also true. (This corresponds to the scenario in when **ϕ** is true, the truth of the implication is the same as the truth of **ψ**.)*

*If you start out with a false premise, then, as far as implication is concerned, you are free to conclude anything. (This corresponds to the scenario in when **ϕ** is false, the implication **ϕ**→**ψ** is true no matter what **ϕ** is.)*

Contradiction

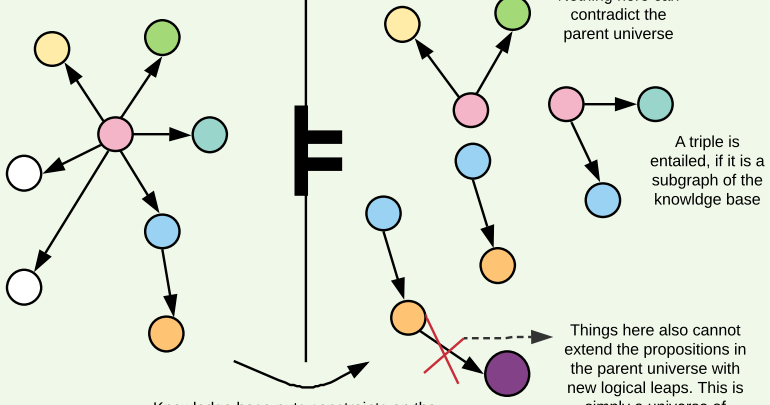
p	q	p→q	¬q	p∧¬q	(p→q)∧(p∧¬q)
T	T	T	F	F	F
T	F	F	T	T	F
F	T	T	F	F	F
F	F	T	T	F	F

ENTAILMENT IN SIMPLE GROUNDED GRAPHS

Models universe consists of all the **subgraphs** of the knowledge base

KNOWLEDGE BASE (Main universe)

MODELS UNIVERSE (Entailments universe, Interpretations universe)



Nothing here can contradict the parent universe

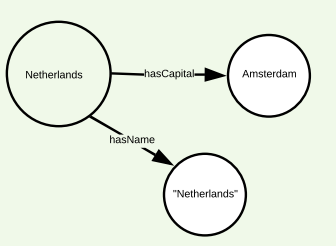
A triple is entailed, if it is a subgraph of the knowldge base

Things here also cannot extend the propositions in the parent universe with new logical leaps. This is simply a universe of permutations for the parent universe.

Knowledge base puts constraints on the models universe; it determines which (sub-graphs) and models are **allowed**; it defines what is posible.

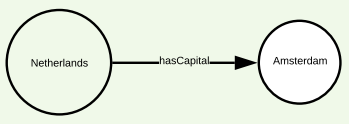
KNOWLEDGE BASE (Main universe)

MODELS UNIVERSE (Entailments universe, Interpretations universe)



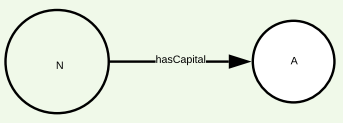
Knowledge Base = {
(Netherlands has_name "Netherlands")
(Netherlands has_capital Amsterdam)
}

Model:



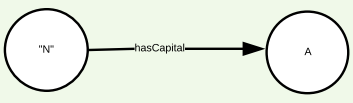
Interpretation(hasCapital)={Netherlands, Amsterdam}

Alternative model:



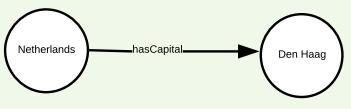
altInterpretation = {@, $, %},
Interpretation(Netherlands) = N
Interpretation("Netherlands") = "N"
Interpretation(Amsterdam) = A
Interpretation(hasCapital)={N, A}

Not a model:



Interpretation(hasCapital)={"Netherlands", Amsterdam}
This statement establishes a relationship between Netherland's name and Amsterdam, and therefore, it is not a model of the knowledge base.

Not a model:



Interpretation(hasCapital)={Netherlands, Den Haag}

This statement establishes a relationship between Netherland's name and Amsterdam, and therefore, it is not a model of the knowledge base.

Image Credits:
Lion by Yu luck from the Noun Project
Animal by Alina Oleynik from the Noun Project
animal footprints by Gagana from the Noun Project