First Order Logic– Arguments (5A)

Young W. Lim
3/10/17

Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using LibreOffice

Based on

Contemporary Artificial Intelligence,
R.E. Neapolitan & X. Jiang

Logic and Its Applications,
Burkey & Foxley

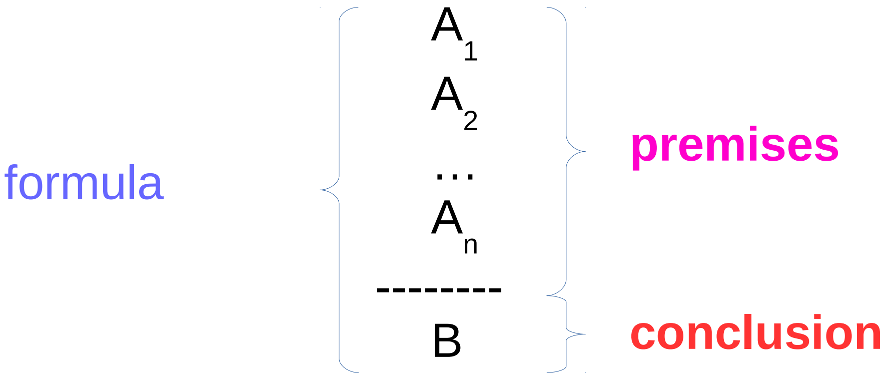
Arguments

An **argument** consists of a set of formula : propositions

The **premises** formula propositions

The **conclusion** formula proposition

List of **premises** followed by the **conclusion**



formula

A_1

A_2

…

A_n

B

premises

conclusion

Formulas and Sentences

An **formula**

- A **atomic formula**
- The operator $\neg$ followed by a **formula**
- Two formulas separated by $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$
- A quantifier following by a variable followed by a formula

A **sentence**

- A **formula** with no free variables

$\forall x$ love(x,y)	: free variable y	: not a sentence
$\forall x$ tall(x)	: no free variable	: a sentence

Entailment Definition

If **the truth of a statement P guarantees** that
another **statement Q** must be **true**,

then we say that **P entails Q**, or that **Q** is **entailed** by **P**.

the key term, “must be true”
“Must be” is stating that
something is **necessary**,
something for which **no other option** or **possibility** exists.
“Must be” encodes the concept of **logical necessity**.

http://www.kslinker.com/VALID-AND-INVALID-ARGUMENTS.html

Entailment Examples

P: “Mary knows all the capitals of the United States”
Q: “Mary knows the capital of Kentucky”

P: “Every one in the race ran the mile in under 5 minutes”
Q: “John, a runner in the race, ran the mile in less than 5 minutes”

P: “The questionnaire had a total of 20 questions, and Mary answered only 13”
Q: “The questionnaire answered by Mary had 7 unanswered questions”

P: “The winning ticket starts with 3 7 9”
Q: “Mary's ticket starts with 3 7 9 so Mary's ticket is the winning ticket”

conceptually familiarity
true **conditional statements**, or true **implications** are examples of **entailment**.

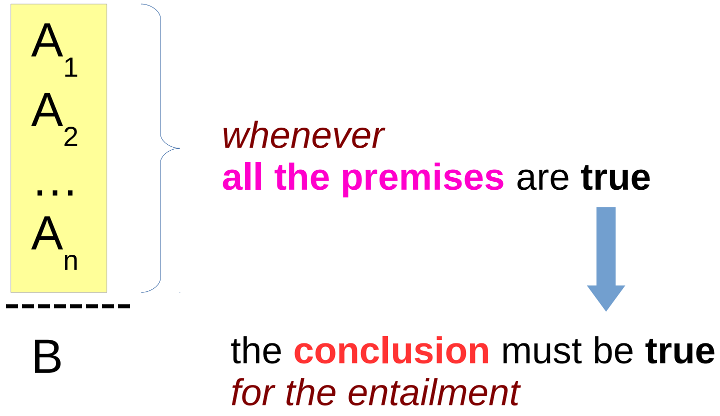
any conditional statement, (“if . . .. then”) which is true, is an example of **entailment**.
In Mathematics, the more familiar term is “**implication**”.

http://www.kslinker.com/VALID-AND-INVALID-ARGUMENTS.html

Entail

The **premises** is said to **entail** the **conclusion**
If in **every model** in which **all the premises** are **true**,
the **conclusion** is also **true**

List of **premises** followed by the **conclusion**



A_1
A_2
…
A_n
--------
B

whenever
all the premises are **true**

the **conclusion** must be **true**
for the entailment

Entailment Notation

Suppose we have *an argument*
whose **premises** are $A_1, A_2, \ldots, A_n$
whose **conclusion** is B

Then

$A_1, A_2, \ldots, A_n \models B$ if and only if
$A_1 \land A_2 \land \ldots \land A_n \Rightarrow B$ (**logical implication**)

logical implication: if $\boxed{A_1 \land A_2 \land \ldots \land A_n \Rightarrow B}$ is tautology (*always true*)

The **premises** is said to **entail** the **conclusion**
If in every model in which
all the premises are **true**,
the **conclusion** is also **true**

Entailment and Logical Implication

$A_1, A_2, \ldots, A_n \models B$

⇔ $A_1 \land A_2 \land \ldots \land A_n \Rightarrow B$

⇔ $A_1 \land A_2 \land \ldots \land A_n \Rightarrow B$ is a tautology

(**logical implication**)

If all the premises are **true**,
then the **conclusion** must be **true**

$T \land T \land \ldots \land T \Rightarrow T$

$T \land T \land \ldots \land T \Rightarrow \cancel{F}$

$F \land X \land \ldots \land X \Rightarrow T$

Sound Argument and Fallacy

A **sound argument**

$A_1, A_2, \ldots, A_n \models B$

If the **premises** **entails** the **conclusion**

A **fallacy**

$A_1, A_2, \ldots, A_n \not\models B$

If the **premises** does **not** **entail** the **conclusion**

Valid Argument Criteria

- If all the premises are true, then the conclusion must be **true**.
- the truth of the conclusion is guaranteed

  if all the premises are **true**
- It is impossible to have a false conclusion

  if all the premises are **true**
- The premises of a valid argument **entail** the conclusion.

http://www.kslinker.com/VALID-AND-INVALID-ARGUMENTS.html

Valid Argument Examples

If John makes this field goal, then the U of A will win.
John makes the field goal .
Therefore the U of A wins
Modus Ponens

If **P** then **Q**
P
Therefore Q

If the patient has malaria, then a blood test will indicate that his blood harbors at least one of these parasites: P. falciparum, P. vivax , P. ovale and P. malaria
Blood test indicate that the patient harbors **none** of these parasites
Therefore the patient does **not** have malaria.
Modus Tollens

If **P** then **Q**
Not **Q**
Therefore Not **P**

Either The Patriots or the Philadelphia Eagles will **win** the Superbowl
The Patriots **lost**
Therefore The Eagles **won**
Disjunctive Syllogism (Process of Elimination)

Either **P** or **Q**
Not **P**
Therefore Q

If John gets a raise, then he will buy a house.
If John buys a house, he will run for a position on the neighborhood council.
Therefore, if John gets a raise, he will run for a position on the neighborhood council
Hypothetical Syllogism

If **P** then **Q**
If **Q** then **R**
Therefore If **P** then **R**

http://www.kslinker.com/VALID-AND-INVALID-ARGUMENTS.html

Valid Arguments

An argument form is **valid** if and only if

whenever the premises are **all true**, then conclusion is **true**.

An argument is valid if its argument form is valid.

If	premises : **true**	→	**then**	conclusion : **true**
	false			**true**
	false			**false**
If	**true**	→	**then never**	**false**

http://math.stackexchange.com/questions/281208/what-is-the-difference-between-a-sound-argument-and-a-valid-argument

Sound Arguments

An argument is **sound** if it is **valid** and all the **premises** are actually **true**.

for an argument to be sound, two conditions must be meet:
1) the argument must be **valid**, and
2) the argument must actually have **all true premises**.

What can be said about the conclusion to a sound argument?

Since the argument is **sound**, then it is both **valid** and actually has **all true premises**, so the **conclusion must be true**, by definition of validity.

an example of a sound argument:

If a number is greater than 7 it is greater than 3.
8 is greater than 7.
Therefore 8 is greater than 3.

http://www.iep.utm.edu/val-snd/

Soundness

A deductive argument is **sound** if and only if

it is both **valid**, and all of its premises are **actually true**.

Otherwise, a deductive argument is **unsound**.

Always premises : **true** ⇨ **therefore** conclusion : **true**

http://www.iep.utm.edu/val-snd/

Models and Interpretations

First specify a **signature**

- Constant Symbols
- Predicate Symbols
- Function Symbols

Determines the language

Given a language

A **model** is specified

A **domain of discourse**

a set of entities

An **interpretation**

- constant assignments
- function assignments
- truth value assignments - predicate

Satisfiability of a sentence

If a sentence s evaluates to **True** under a given interpretation I

I satisfies s; **I** ⊨ **s**

A sentence is **satisfiable**
if there is some interpretation under which it is **true**.

Valid Formulas and Sentences

A formula is **valid**
if it is **satisfied** by ***every*** interpretation

Every **tautology** is a **valid** formula

A **valid** sentence: $\text{human(John)} \lor \neg\text{human(John)}$

A **valid** sentence: $\exists x\ (\text{human}(x) \lor \neg\text{human}(x))$

A **valid** formula: $\text{loves(John}, y) \lor \neg\text{loves(John}, y)$

True regardless of which individual
in the domain of discourse is assigned to y
This formula is true in every interpretation

Validity and Satisfiability of Formulas

A formula is **valid** if it is **true** for all values of its terms.

Satisfiability refers to the existence of a combination of values to make the expression **true**.

So in short, a proposition / a formula is

- **satisfiable** if there is at least one **true** result in its truth table,
- **valid** if all values it returns in the truth table are **true**.

Contradiction

A sentence is a **contradiction** if there is **no** interpretation that satisfies it

$\exists x \, (\text{human}(x) \wedge \neg \text{human}(x))$

not satisfiable under any interpretation

Well-formed Strings

A **term**

- A constant symbol
- A variable symbol
- A function symbol with comma separated list

An **atomic formula**

- A predicate symbol
- A predicate symbol with comma separated list
- Two terms separated by the = symbol

An **formula**

- A **atomic formula**
- The operator ¬ followed by a formula
- Two formulas separated by ∧, ∨, ⇒, ⇔
- A quantifier following by a variable followed by a formula

A sentence

- A **formula** with no free variables

Arguments

An argument consists of set of formulas, called the premises
And a formula called the conclusion

The premises **entail** the conclusion
If *in every model* in which all the premises are **true**,
the conclusion is also **true**.

If the premises **entail** the conclusion,
the argument is **sound**
otherwise, it is a **fallacy**

A **set** of **inference rules** : a deductive system
A deductive system is **sound** if it only derives sound arguments
A deductive system is **complete** if it can derive every sound argument

Quantifiers

Universal Quantifier

$\forall x$ for all entity *e* in the *domain of discourse*

Existential Quantifier

$\exists x$ for some entity *e* in the *domain of discourse*

Instantiation and Generalization

Universal Instantiation

$\forall x\ A(x)$ for all entities e in the *domain of discourse* $\models$ $A(c)$ for any *constant* term c

Universal Generalization

$A(e)$ for every entity e in the *domain of discourse* $\models$ $\forall x\ A(x)$

Existential Instantiation

$\exists x\ A(x)$ $\models$ $A(e)$ for some entity e in the *domain of discourse*

Existential Generalization

$A(e)$ for an entity e in the *domain of discourse* $\models$ $\exists x\ A(x)$

Universal Instantiation

$\forall x\ A(x)$ for <u>all entities</u> x in the *domain of discourse* $\models$ $A(c)$ for any *constant* term c

If A(*x*) has value **T** for <u>all entities</u> in the domain of discourse,
then it must have value **T** for <u>term *t*</u>

man(John)
$\forall x$ man(x) ⇒ human(x)
human(John)

man(John)
$\forall x$ man(x) ⇒ human(x)
man(*John*) ⇒ human(*John*)
human(*John*)

Universal Generalization

$A(e)$ for every entity e in the *domain of discourse* $\models \forall x\, A(x)$

If $A(e)$ has value **T** for every entity e,
Then $\forall x\, A(x)$ has value **T**

The rule is ordinarily applied by showing that $A(e)$ has value **T**
for an arbitrary entity e

$\forall x\, (\text{man}(x) \Rightarrow \text{human}(x))$
$\forall x\, (\neg\text{human}(x) \Rightarrow \neg\text{man}(x))$

$\forall x\, (\text{man}(x) \Rightarrow \text{human}(x))$
$\text{man}(e) \Rightarrow \text{human}(e)$
$\neg\text{human}(e)$
$\neg\text{man}(e)$
$\neg\text{human}(e) \Rightarrow \neg\text{man}(e)$
$\forall x\, (\neg\text{human}(x) \Rightarrow \neg\text{man}(x))$

Modus Tolens

$A \Rightarrow B$
$\neg B$
$\neg A$

Existential Instantiation

$\exists x\ A(x) \models A(e)$ for some entity e in the *domain of discourse*

If $\exists$x A(x) has value **T**,
then A(e) has value **T** for some entity e

$\exists$x man(x)
$\forall x$ man(x) $\Rightarrow$ human(x)
$\exists$x human(x)

$\exists$x man(x)
$\forall x$ man(x) $\Rightarrow$ human(x)
man(e)
man(e) $\Rightarrow$ human(e)
human(e)
$\exists$x human(x)

Existential Generalization

$A(e)$ for <u>an entity</u> e in the *domain of discourse* $\models \exists x\, A(x)$

If A(e) has value T for some entity e,
Then ∃x A(x) has value T

man(John)
$\forall x$ man(x) ⇒ human(x)
∃x human(x)

man(John)
$\forall x$ man(x) ⇒ human(x)
man(*John*) ⇒ human(*John*)
human(*John*)
∃x human(x)

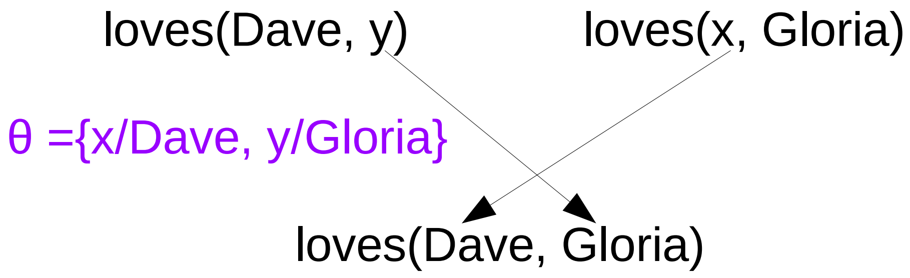
Unification

Two sentences **A** and **B**

A **unification** of **A** and **B**

A substitution θ of values for some of the **variables** in **A** and **B**
that make the sentences *identical*

The **set** of substitutions θ is called the **unifier**

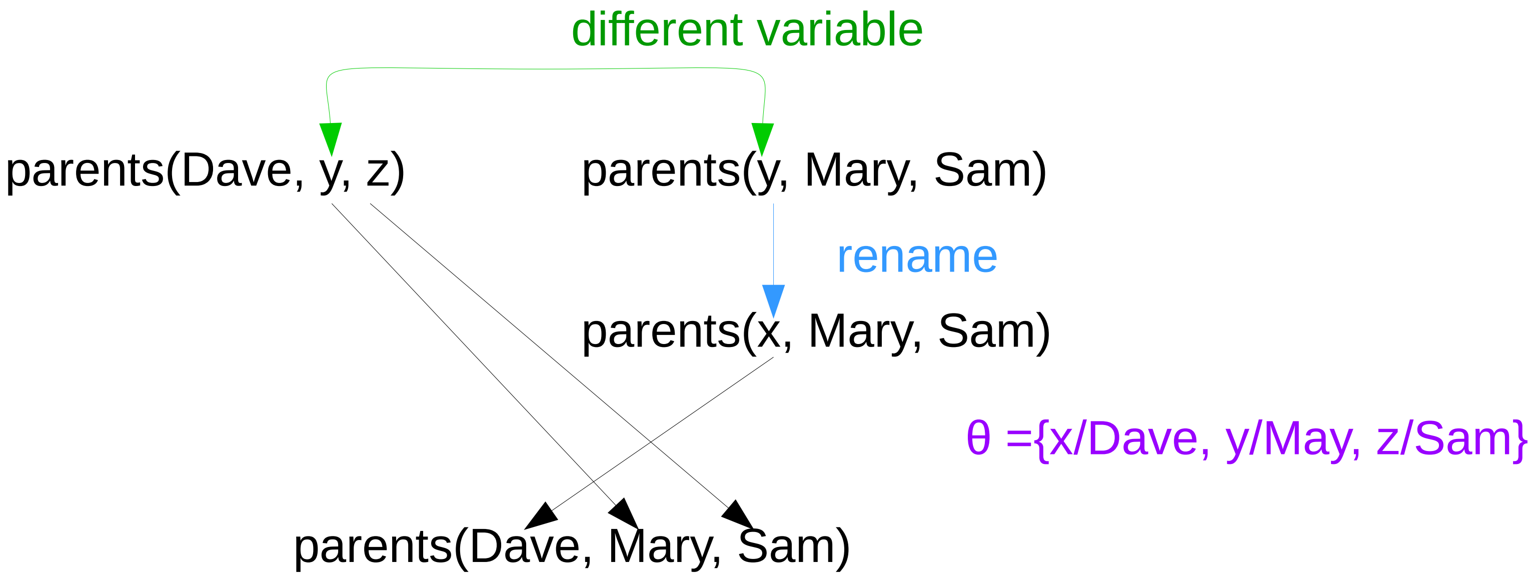


loves(Dave, y) loves(x, Gloria)

θ ={x/Dave, y/Gloria}

loves(Dave, Gloria)

Unification



different variable

parents(Dave, y, z)

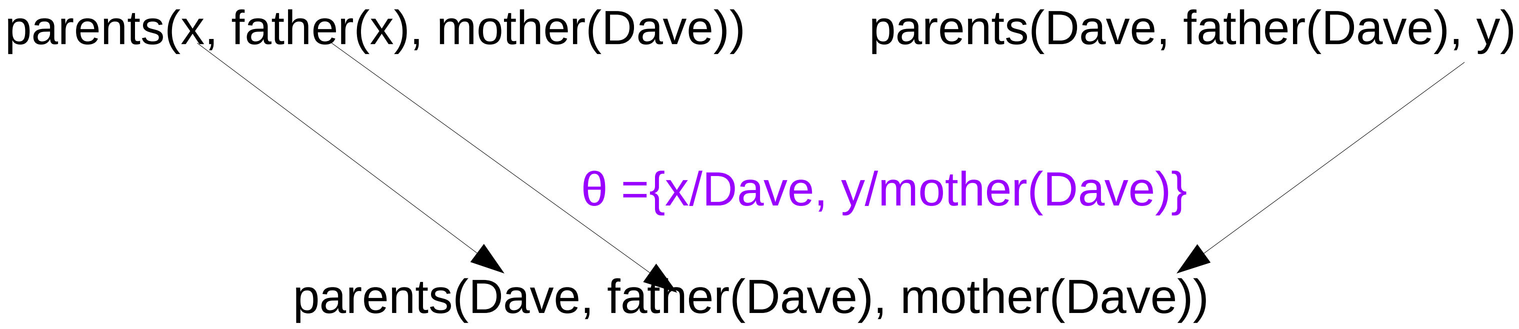
parents(y, Mary, Sam)

rename

parents(x, Mary, Sam)

θ ={x/Dave, y/May, z/Sam}

parents(Dave, Mary, Sam)



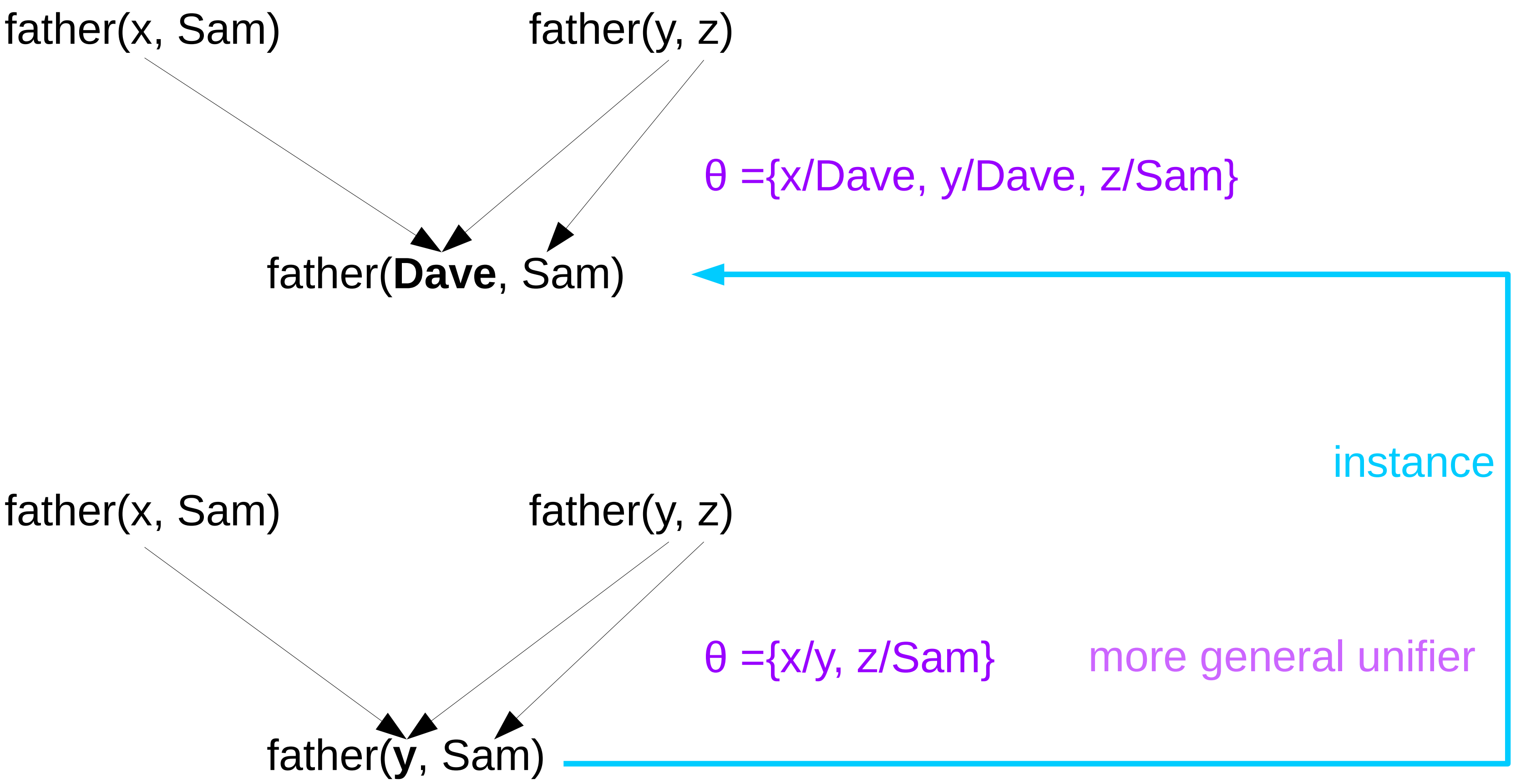
parents(x, father(x), mother(Dave))

parents(Dave, father(Dave), y)

θ ={x/Dave, y/mother(Dave)}

parents(Dave, father(Dave), mother(Dave))

Unifier Instance



father(x, Sam)

father(y, z)

θ ={x/Dave, y/Dave, z/Sam}

father(**Dave**, Sam)

instance

father(x, Sam)

father(y, z)

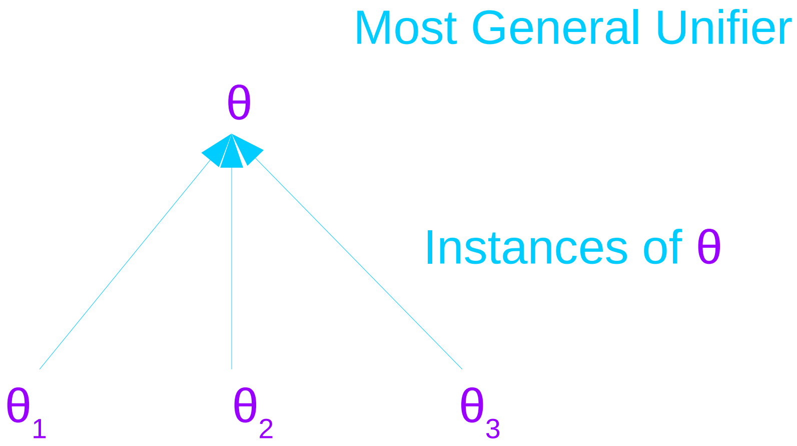
θ ={x/y, z/Sam}

more general unifier

father(**y**, Sam)

Most General Unifier

If every other unifier θ’ is
an instance of θ
in the sense that θ’ can be derived
by making substitutions in θ



Most General Unifier

θ

Instances of θ

θ_1 θ_2 θ_3

father(**y**, Sam)

y/Dave

father(**Dave**, Sam

Unification Algorithm

Input : Two sentences A and B; an empty set of substitution theta
Output : a most general unifier of the sentences if they can be unified;
otherwise failure.

```
Procedure unify (A, B, var theta)
Scan A and B from left to right
   Until A and B disagree on a symbol or A and B are exhausted
If A and B are exhausted
  Let x and y be the symbols where A and B disagree
  If x is a variable
    Theta = theta U {x/y}
    unify(subst(theta, A), subst(theta, B), theta)
  Else if y is a variable
    Theta = theta U {y/x}
    unify(subst(theta, A), subst(theta, B), theta)
  Else
    Theta = failure;
  Endif
endif
```

Subst

Procedure subst
Input : a set of substitutions θ and a sentence A
Applies the substitutions θ in to A

A: parents(Dave, y, z)
θ = {x/Dave, y/Mary, z/Sam}

subst(A, θ)

parents(Dave, May, Sam)

Generalized Modus Ponens

Suppose we have sentences A, B, and C, and the the sentence A $\Rightarrow$ B, which is implicitly universally quantified for all variables in the sentence.

The Generalized Modus Ponens (GMP) rule is as follows

$A \Rightarrow B, C, \text{unify}(A, C, \theta) \models \text{subst}(B, \theta)$

Modus Ponens

1. mother(Mary, Scott)
2. sister(Mary, Alice)
3. $\forall x \forall y \forall z\ \text{mother}(x,y) \land \text{sister}(x,z) \Rightarrow \text{aunt}(z,y)$

4. $\text{mother}(\text{Mary, Scott}) \land \text{sister}(\text{Mary, Alic})$
5. $\forall y \forall z\ \text{mother}(\text{Mary},y) \land \text{sister}(\text{Mary},z) \Rightarrow \text{aunt}(z,y)$
6. $\forall z\ \text{mother}(\text{Mary},\text{Scott}) \land \text{sister}(\text{Mary},z) \Rightarrow \text{aunt}(z,\text{Scott})$
7. $\text{mother}(\text{Mary},\text{Scott}) \land \text{sister}(\text{Mary},\text{Alice}) \Rightarrow \text{aunt}(\text{Alice},\text{Scott})$
8. aunt(Alice,Scott)

Generalized Modus Ponens

1. mother(Mary, Scott)
2. sister(Mary, Alice)
3. $\forall x \forall y \forall z$ mother(x,y) $\wedge$ sister(x,z) $\Rightarrow$ aunt(z,y)

4. mother(Mary, Scott) $\wedge$ sister(Mary, Alic)
5. θ = {x/Mary, y/Scott, z/Alice}
6. aunt(Alice,Scott)

Logical Equivalences

¬, ∧ ,
∨

¬, ∧ ,
∨

∧∨⌐¬⇒
⇔≡⇒ ⊨

∧∨⌐¬⇒
⇔≡⇒⊨

⇒
⇔
≡

⇒
⇔
≡

References

[1] en.wikipedia.org
[2] en.wiktionary.org
[3] U. Endriss, “Lecture Notes : Introduction to Prolog Programming”
[4] http://www.learnprolognow.org/ Learn Prolog Now!
[5] http://www.csupomona.edu/~jrfisher/www/prolog_tutorial
[6] www.cse.unsw.edu.au/~billw/cs9414/notes/prolog/intro.html
[7] www.cse.unsw.edu.au/~billw/dictionaries/prolog/negation.html
[8] http://ilppp.cs.lth.se/, P. Nugues,` An Intro to Lang Processing with Perl and Prolog