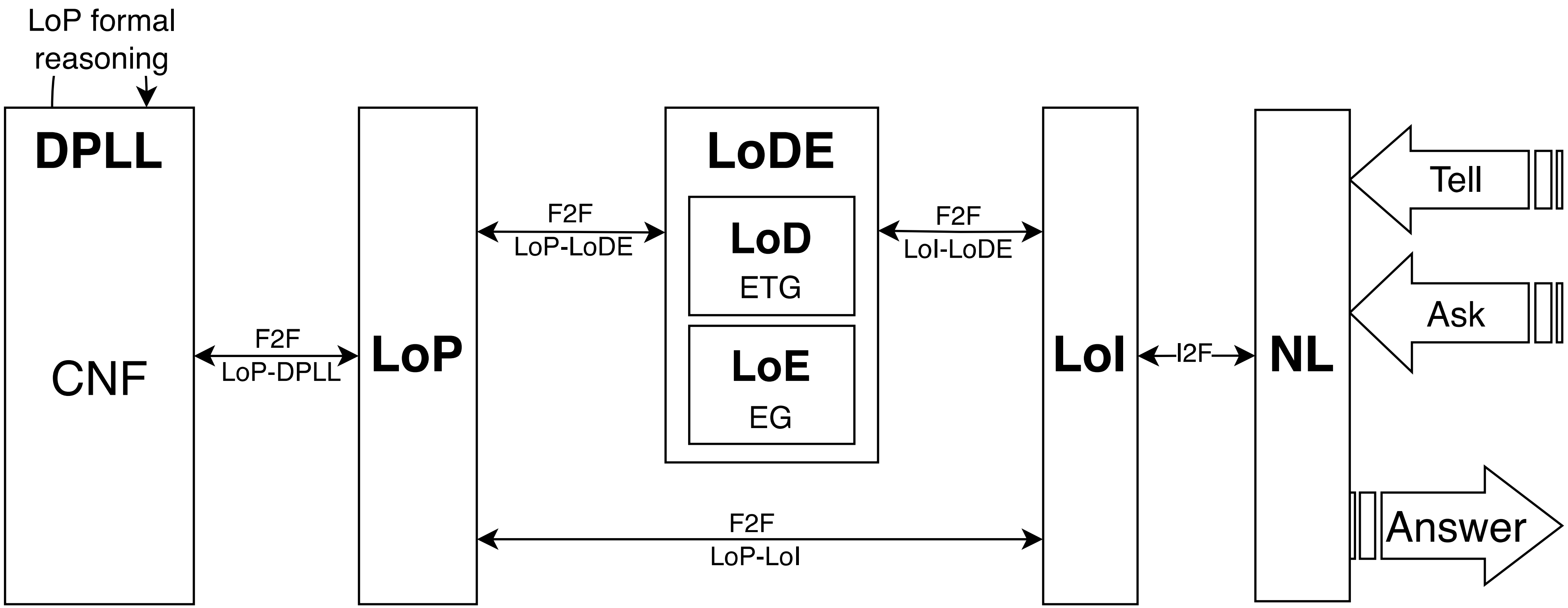
LoP- The Logic of Propositions Reasoning

(T2MP)

Reasoning about propositions

- **Intuition**
- NL to LoI
- LoI to ALC / LoDE
- LoI to LoP
- LoP to CNF
- CNF reasoning
- Reasoning about propositions – end-to-end

Reasoning about propositions – the big picture



Notation. NL: Natural (informal) Language. I2F: Informal to Formal. F2F: Formal to Formal

Reasoning about propositions

- Intuition
- **NL to LoI**
- ALC / LoDE to LoI
- LoI to LoP
- LoP to CNF
- CNF reasoning
- Reasoning about propositions – end-to-end

Implication and universal quantification

Observation (⊃ and ∀). The scope of a universal quantification can be weakened by making it the consequence of an implication where the premise defines the scope.

Example (Correct use of ⊃ and ∀). The sentence

"Everybody working at UniTn is smart"

should be translated as the formula

$$\forall x.\ (WorksAt(UniTn, x) \supset Smart(x))$$

where the implication limits the extent of the universal quantification.

Example (Wrong use of ∧ and ∀). The mistake is to use a conjunction, which not only does not restrict the scope but it strengthens the statement. Infact, the formula

$$\forall x.\ (WorksAt(UniTn, x) \land Smart(x))$$

means

"Everybody works at UniTn and eveybody is smart"

Implication and universal quantification – example

Example ($\wedge$, $\supset$ and $\forall$). Consider the following sentence

"Those students who study and are smart"

What is the right translation?

Solution. We have the following:

Wrong translation:

$$\forall x.(Student(x) \supset Smart(x))$$

Correct translation:

$$\forall x.(Student(x) \wedge Smart(x))$$

The wrong translation translates the sentence:

"Those students who study are smart"

Conjunction and existential quantification

Observation (∧ and ∃). The scope of an existential quantification can be weakened by adding a conjunct which adds an extra constraint to be satisfied thus making the existential quantification stronger (more constraints to be satisfies.

Example (Correct use of ∧ and ∃). The sentence

"There is a person working at UniTn and she is smart"

should be translated as he formula

$$\exists x. (WorksAt(UniTn, x) \wedge Smart(x))$$

Example (Wrong use of ⊃ and ∃). The mistake is to use an implication (that is, a disjunction) which weakens, rather than strengthening, the existential quantification. Infact, the formula

$$\exists x. (WorksAt(UniTn, x) \supset Smart(x))$$

means

"There is a person so that if she works at UniTn then she is smart"

Conjunction and existential quantification – example

Example (∧ , ⊃ and ∀). Consider the following sentence

"There is a someone who is smart because it is a student"

What is the right translation?

Solution. We have the following:

Wrong translation:

$$\exists x.(Student(x) \wedge Smart(x))$$

Correct translation:

$$\exists x.(Student(x) \supset Smart(x))$$

The wrong translation translates the sentence:

"There is an intelligent student "

Translation of "There are *at least n*"

Observation (Translation of "There are *at least* 1"). Existential quantification establishes that there is *at least* one element of the domain satisfying the property in its scope. It does not allow to put a lower bound in the number of domain elements satisfying the property in its scope.

Example (Wrong translation of "There are *at least 2*"). The formula

$$\exists x_1.\exists x_2.(attend\,(x_1, Logic) \wedge attend\,(x_2, Logic))$$

Is not saying that

"there are at least two students attending the Logic class".

The above representation is not enough, as x_1 and x_2 could denote the same individual. In order to enforce a minimal number, we have to guarantee that x_1 and x_2 denote different indivudals.

Translation of "There are *at least n*" – continued

Example (Correct translation of "There are *at least 2*") The formula saying that

"there are at least two students attending the Logic class"

is as follows

$$\exists x_1.\exists x_2.(attend(x_1, Logic) \wedge attend(x_2, Logic) \wedge x_1 \neq x_2)$$

Observation (Correct translation of "There are *at least n*") In order to guarantee a minimal number of elements we need to enforce that the different existential quantifications insist on different elements. Let ϕ be the formula that we want to be satisfied by at least n elements. The we have the following:

$$\exists x_1 \ldots x_n.(\bigwedge_{i=1}^{n} \phi(x_i) \wedge \bigwedge_{i \neq j = 1}^{n} x_j \neq x_i)$$

Translation of "There are *at most n*"

Observation (Translation of "There are at most *n*"). Universal quantification establishes that all the elements of the domain satisfy the property in its scope. It does not allow to put an upper bound in the number of domain elements satisfying the property in its scope.

Example (Wrong translation of "There are at most *2*"). Write the formula saying that *"there are at most two students attending the Logic class"*.

$$\forall x_1.\forall x_2 . (attend(x_1, Logic) \lor attend(x_2, Logic))$$

The above representation is not enough, as x_1 and x_2 could denote different individuals. In order to enforce a minimal number, we have to guarantee that x_1 and x_2 denote the same two people.

Translation of "There are *at most n*" – continued

Example (Correct translation of "There are at most *2*") Write the formula saying that

"there are at most two students attending the Logic class"

is as follows:

$$\forall x_1.\forall x_2.\forall x_3.(attend(x_1, Logic) \wedge attend(x_2, Logic) \wedge attend(x_3, Logic)) \supset (x_1 = x_2 \vee x_2 = x_3 \vee x_1 = x_3)$$

Observation (Correct translation of "There are at most *2*"). In order to guarantee a maximal number of elements we enforce that the different universal quantifications insist on the same elements. Let ϕ be the formula to be satisfied by at most n elements:

$$\forall x_1 \ldots x_n.(\bigwedge_{i=1}^{n} \phi(x_i) \supset \bigvee_{i \neq j = 1}^{n} x_j = x_i)$$

Exercise

Exercise. Formalizzare in LoI la seguente sentenza

Ci sono due studenti intelligenti

Exercise. Formalizzare in LoI la seguente sentenza

Ci sono non più di due studenti che disturbano

Exercise. Can you express something along the lines of:

La maggior parte degli studenti è intelligente?

Simple answer: no.

Complex answer: fix the domain and know how many different constants you have, you can write a formula which expands the condition of uniqueness up to the required number of constants (size / 2 + 1)

Exercise

Exercise. Translate The following LoI formulas to natural language

1. $\exists x.(bought(Frank, x) \wedge dvd(x))$
2. $\exists x.bought(Frank, x)$
3. $\forall x.(bought(Frank, x) \supset bought(Susan, x))$
4. $(\forall x.bought(Frank, x)) \supset (\forall x.bought(Susan, x))$
5. $\forall x.\exists y.bought(x, y)$
6. $\exists x.\forall y.bought(x, y)$

Exercise

Exercise. Translate the following natural language sentences into LoI

1. Every man is mortal
2. Every dog has a tail
3. There are two dogs
4. Not every dog is white
5. There is dog
6. There is at most one dog

Exercise

Exercise. Define an appropriate language (constants, functional and relational symbols) and formalize the following sentences using LoI formulas.

1. All students are smart
2. There exists a student
3. There exists a smart student
4. Every student loves some student
5. Every student loves some other student
6. There is a student who is loved by every other student
7. Bill is a student
8. Bill takes either Analysis or Geometry (but not both)
9. Bill takes Analysis and Geometry
10. Bill doesn't take Analysis
11. No student loves Bill

Exercise

Exercise. Given the formula

$$adult(x) \supset contains(document(x), photo(x))$$

indicate which of the following statements are true:

1. adult is a functional symbol, contains is a predicative symbol, document is a functional symbol, photo is a functional symbol, x is a variable
2. adult is a predicative symbol, contains is a predicative symbol, document(x) is a term, photo is a functional symbol, x is a variable
3. adult, contains, document and photo are all functional symbols, x is a variable
4. adult, contains, document, and photo are all predicate symbols, x is a term
5. adult is a functional symbol, contains(x) it's a functional symbol, document is a predicative symbol, photo is a predicative symbol, x is a functional symbol

Exercise

Exercise. Given the sentence "Every red mushroom is poisonous", and using the following predicative symbols:

- Red(x), to say that x is red
- Mushroom(x), to say that x is a mushroom
- Poisonous(x), to say that x is poisonous

Which of the following formalization represent correctly the informal sentence above?

1. $\exists x.(Red(x) \wedge Mushroom(x) \supset Poisonous(x))$
2. $\forall x.(Mushroom(x) \wedge Red(x)) \supset \exists x.Poisonous(x)$
3. $\forall x.(Poisonous(x) \supset (Red(x) \wedge Mushroom(x)))$
4. $\forall x.(Red(x) \supset (Mushroom(x) \supset Poisonous(x)))$
5. $\forall x.(Red(x) \wedge Mushroom(x) \wedge Poisonous(x))$
6. $\forall x.(Red(x) \wedge Mushroom(x) \supset Poisonous(x))$

Exercise

Exercise. Formalize in FOL the following sentence:

There is only one student who failed the Geometry exam.

1. $\exists x.(Student(x) \wedge Failed(x, Geometry) \wedge \forall y.(Student(y) \wedge Failed(y, Geometry) \supset x = y))$
2. $\forall x.(Student(x) \wedge Failed(x, Geometry) \wedge \exists y.(Student(y) \wedge Failed(y, Geometry) \supset x = y))$
3. $\forall x.(Student(x) \supset Failed(x, Geometry) \wedge \forall y.(Student(y) \wedge Failed(y, Geometry) \supset x = y))$
4. $\exists x.\exists y.(Student(x) \wedge Failed(x, Geometry) \wedge x = y)$

Reasoning about propositions

- Intuition
- NL to LoI
- **ALC / LoDE to LoI**
- LoI to LoP
- LoP to CNF
- CNF reasoning
- Reasoning about propositions – end-to-end

ALC / LoDE to LoI

- Cyclist ≡ Person ⊓ ∃rides.Bike

  ∀x.(Cyclist(x) ≡ Person(x) ∧ ∃y.(rides(x,y) ∧ Bike(y)))

- Runner ≡ Person ⊓ ∃likesTo.Run

  ∀x.(Runner(x) ≡ Person(x) ∧ ∃y.(likesTo(x,y) ∧ Run(y)))

- Bike ⊑ Vehicle ⊓ ∀actuator.Pedals

  ∀x.(Bike(x) ⊃ (Vehicle(x) ∧ ∀y.(actuator(x,y) ⊃ Pedals(y))))

- Cyclist(Mike)

  Cyclist(Mike)

Reasoning about propositions

- Intuition
- NL to LoI
- LoI to ALC / LoDE
- **LoI to LoP**
- LoP to CNF
- CNF reasoning
- Reasoning about propositions – end-to-end

LoI to LoP

$\forall x.(Cyclist(x) \supset \exists y.(rides(x,y) \wedge Bike(y)))$ D={Mike, Bike#1}

$Cyclist(Mike) \supset ((rides(Mike, Bike\#1) \wedge Bike(Bike\#1)) \vee$

$(rides(Mike, Mike) \wedge Bike(Mike))$

$\wedge\ Cyclist(Bike\#1) \supset ((rides(Bike\#1, Bike\#1) \wedge Bike(Bike\#1)) \vee$

$(rides(Bike\#1, Mike) \wedge Bike(Mike))$

Reasoning about propositions

- Intuition
- NL to LoI
- ALC / LoDE to LoI
- LoI to LoP
- **LoP to CNF**
- CNF reasoning
- Reasoning about propositions – end-to-end

Conversion to CNF (1)

Definition 4 (The CNF function) Given a PL formula ϕ the function CNF, which transforms ϕ in its CNF form, called CNF(ϕ) is recursively defined as follows:

$CNF(p)$	$=$	p if $p \in$ **PROP**
$CNF(\neg p)$	$=$	$\neg p$ if $p \in$ **PROP**
$CNF(\phi \supset \psi)$	$=$	$CNF(\neg\phi) \otimes CNF(\psi)$
$CNF(\phi \wedge \psi)$	$=$	$CNF(\phi) \wedge CNF(\psi)$
$CNF(\phi \vee \psi)$	$=$	$CNF(\phi) \otimes CNF(\psi)$
$CNF(\phi \equiv \psi)$	$=$	$CNF(\phi \supset \psi) \wedge CNF(\psi \supset \phi)$
$CNF(\neg\neg\phi)$	$=$	$CNF(\phi)$
$CNF(\neg(\phi \supset \psi))$	$=$	$CNF(\phi) \wedge CNF(\neg\psi)$
$CNF(\neg(\phi \wedge \psi))$	$=$	$CNF(\neg\phi) \otimes CNF(\neg\psi)$
$CNF(\neg(\phi \vee \psi))$	$=$	$CNF(\neg\phi) \wedge CNF(\neg\psi)$
$CNF(\neg(\phi \equiv \psi))$	$=$	$CNF(\phi \wedge \neg\psi) \otimes CNF(\psi \wedge \neg\phi)$

... see next page

Conversion to CNF (2 – continued)

… where

$$(C_1 \land \ldots \land C_n) \otimes (D_1 \land \ldots \land D_m) \qquad (*)$$

is defined as:

$$(C_1 \lor D_1) \land \ldots \land (C_1 \lor D_m) \land \ldots \land (C_n \lor D_1) \land \ldots \land (C_n \lor D_m) \qquad (**)$$

with C_i being a conjunction (possibly a single formula) and D_j being a disjunction (possibly a single formula).

Example (special cases). Rewrite the following special cases of (*) into their corresponding formulas (**)

- Single formula conjuncts: $(a \land \mathrm{b}) \otimes (D_1 \land \ldots \land D_m)$
- Single formula disjuncts: $(C_1 \land \ldots \land C_n) \otimes (a \land \mathrm{b})$
- Single formula conjuncts and disjuncts: $(a \land \mathrm{b}) \otimes (c \land \mathrm{d})$

Conversion to CNF (example)

Example (CNF conversion). Compute the CNF of $(p \supset q) \equiv (\neg q \supset \neg p)$

$$\mathrm{CNF}(\,(p \supset q) \equiv (\neg q \supset \neg p)\,) =$$

$$\mathrm{CNF}(\,(p \supset q) \supset (\neg q \supset \neg p)\,) \wedge \mathrm{CNF}(\,(\neg q \supset \neg p) \supset (p \supset q)\,) =$$

$$\mathrm{CNF}(\,\neg(p \supset q)\,) \otimes \mathrm{CNF}(\,\neg q \supset \neg p\,) \wedge \mathrm{CNF}(\,\neg(\neg q \supset \neg p)\,) \otimes \mathrm{CNF}(\,p \supset q\,) =$$

$$(\mathrm{CNF}(\,p\,) \wedge \mathrm{CNF}(\,\neg q\,)) \otimes (\mathrm{CNF}(\,q\,) \otimes \mathrm{CNF}(\,\neg p\,)) \wedge (\mathrm{CNF}(\,\neg q\,) \wedge \mathrm{CNF}(\,p\,)) \otimes (\mathrm{CNF}(\,\neg p\,) \otimes \mathrm{CNF}(\,q\,)) =$$

$$(p \wedge \neg q) \otimes (q \otimes \neg p) \wedge (\neg q \wedge p) \otimes (\neg p \otimes q) =$$

$$(p \wedge \neg q) \otimes (q \vee \neg p) \wedge (\neg q \wedge p) \otimes (\neg p \vee q) =$$

$$(p \vee q \vee \neg p) \wedge (\neg q \vee q \vee \neg p) \wedge (\neg q \vee q \vee \neg p) \wedge (\neg p \vee q \vee p)$$

Conversion to CNF (example)

Example (CNF conversion). Compute the CNF of $(p \wedge r) \supset q$

$$CNF((p \wedge r) \supset q) =$$
$$CNF((\neg(p \wedge r))) \otimes CNF(q) =$$
$$(CNF((\neg p)) \otimes CNF(\neg r)) \otimes q =$$
$$(\neg p \otimes \neg r) \otimes q =$$
$$(\neg p \vee \neg r) \otimes q =$$
$$(\neg p \vee q) \vee (\neg r \vee q) =$$
$$\neg p \vee q \vee \neg r$$

Conversion to CNF (example)

Example (single conjunct and single disjunct)

$$CNF(\ (a \land b) \lor \neg(c \supset d)\ ) =$$
$$CNF(a \land b) \otimes CNF(\neg(c \supset d)) =$$
$$(CNF(a) \land CNF(b)) \otimes (CNF(c) \land CNF(\neg d)) =$$
$$(a \land b) \otimes (c \land \neg d) =$$
$$(a \lor c) \land (a \lor \neg d) \land (b \lor c) \land (b \lor \neg d) =$$

CNF conversion (example)

Example (Exponential explosion of a CNF conversion). Try computing the CNF of the following formula

$$p1 \equiv (p2 \equiv (p3 \equiv (p4 \equiv (p5 \equiv p6)))).$$

The formula resulting from the first conversion step is:

$$CNF(p1 \supset (p2 \equiv (p3 \equiv (p4 \equiv (p5 \equiv p6))))) \land CNF((p2 \equiv (p3 \equiv (p4(p5 \equiv p6)))) \supset p1)$$

This formula is double the length of the previous formula. Continuing the expansion, the formula will keep growing exponentially.

CNF conversion (example)

CNF(*p*1 ⊃ (*p*2 ≡ (*p*3 ≡ (*p*4 ≡ (*p*5 ≡ *p*6))))) ∧ *CNF*((*p*2 ≡ (*p*3 ≡ (*p*4 ≡(*p*5 ≡ *p*6)))) ⊃ *p*1)

CNF(¬p1) ⊗ CNF(*p*2 ≡ (*p*3 ≡ (*p*4 ≡ (*p*5 ≡ *p*6)))) ∧ CNF(¬(*p*2 ≡ (*p*3 ≡ (*p*4 ≡(*p*5 ≡ *p*6))) ⊗ CNF(p1)

¬p1 ⊗ CNF(p2 ⊃ (*p*3 ≡ (*p*4 ≡ (*p*5 ≡ *p*6)))) ∧ CNF((*p*3 ≡ (*p*4 ≡ (*p*5 ≡ *p*6))) ⊃ p2) ∧ CNF(p2 ∧ ¬(*p*3 ≡ (*p*4 ≡(*p*5 ≡ *p*6)))) ⊗ CNF(¬p2 ∧ (*p*3 ≡ (*p*4 ≡(*p*5 ≡ *p*6)))) ⊗ p1

¬p1 ⊗ CNF(¬p2) ⊗ CNF(*p*3 ≡ (*p*4 ≡ (*p*5 ≡ *p*6))) ∧ CNF(¬ (*p*3 ≡ (*p*4 ≡ (*p*5 ≡ *p*6)))) ⊗ CNF(p2) ∧ CNF(p2) ∧ CNF(¬(*p*3 ≡ (*p*4 ≡(*p*5 ≡ *p*6)))) ⊗ CNF(¬p2) ∧ CNF(*p*3 ≡ (*p*4 ≡(*p*5 ≡ *p*6)))) ⊗ p1

…

Reasoning about propositions

- Intuition
- NL to LoI
- LoI to ALC / LoDE
- LoI to LoP
- LoP to CNF
- **CNF reasoning**
- Reasoning about propositions – end-to-end

CNF satisfiability

Proposition (CNF satisfiability). Let ϕ be a formula with

$$CNF(\phi) = C_0, \ldots C_n$$

where $C_0, \ldots, C_n$ are the clauses in CNF(ϕ). Let us assume that we iterate the process of literal evaluation.
Then, the process will terminate with one of two possible situations:

- {}, that is, with an **empty set of clauses**, in which case ϕ is satisfiable;
- {…{}…}, that is, with a **non empty set of clauses containing one empty clause**, in which case ϕ is unsatisfiable.

CNF satisfiability (example)

Example. Check the satisfiability of the following formula

$$(p \lor q) \land (p \lor \neg p) \land (\neg q \lor q) \land (\neg q \lor \neg p) \land (\neg q \lor \neg p) \land (\neg q \lor \neg p) \land (\neg q \lor q) \land (p \lor \neg p) \land (p \lor q)$$

1. $(p \lor q) \land (p \lor \neg p) \land (\neg q \lor q) \land (\neg q \lor \neg p) \land (\neg q \lor \neg p) \land (\neg q \lor \neg p) \land (\neg q \lor q) \land (p \lor \neg p) \land (p \lor q)$
2. $\{\{p, q\}, \{p, \neg p\}, \{\neg q, q\}, \{\neg q, \neg p\}, \{\neg q, \neg p\}, \{\neg q, \neg p\}, \{\neg q, q\}, \{p, \neg p\}, \{p, q\}\}$
3. $\{\{\top, q\}, \{\top, \bot\}, \{\neg q, q\}, \{\neg q, \bot\}, \{\neg q, \bot\}, \{\neg q, \bot\}, \{\neg q, q\}, \{\top, \bot\}, \{\top, q\}\} |_p$
4. $\{\{\neg q, q\}, \{\neg q\}, \{\neg q\}, \{\neg q\}, \{\neg q, q\}\} |_{\neg q}$
5. $\{\{\top, \bot\}, \{\top\}, \{\top\}, \{\top\}, \{\top, \bot\}\}$
6. $\{\}$

CNF satisfiability (example)

Example. Check the satisfiability of the following formula

$$(q \lor \neg p) \land (q \lor \neg p)$$

1. $(q \lor \neg p) \land (q \lor \neg p)$
2. $\{\{q, \neg p\}, \{q, \neg p\}\}|_{\neg p}$
3. $\{\{q, \top\}, \{q, \top\}\}$
4. $\{\}$

CNF Model checking (example)

Example. Select whether the interpretation which makes p and q True is a model for the formula below (which is satisfiable):

$$(q \vee \neg p) \wedge (\neg q \vee p) \wedge (p \vee q)$$

1. $(q \vee \neg p) \wedge (\neg q \vee p) \wedge (p \vee q)$
2. $\{\{q, \neg p\}, \{\neg q, p\}, \{p, q\}\}$
3. $\{\{q, \top\}, \{\neg q, \bot\}, \{\bot, q\}\}|_{\neg p}$
4. $\{\{\neg q\}, \{q\}\}|_{q}$
5. $\{\{\bot\}, \{\top\}\}$
6. $\{\{\}\}$

DPLL decision procedure – base

Algorithm DPLL

Input: $\varphi = \{c_1, \ldots, c_n\}$.

Output: I.

Call DPLL(φ, {})

DPLL(φ, I)

if {} $\in \varphi$

    then exit-return {} **end**

if φ = {}

    then return I **end**

L← select-literal(φ);

DPLL($\varphi|_L$, $I \cup \{L\}$) **or DPLL**($\varphi|_{\neg L}$, $I \cup \{\neg L\}$)

Unit propagation – enhancement 1

Observation (Unit propagation).). Assume that we have a unit clause in the input formula. How would you modify the algorithm produced in the previous step to take into account this situation. When do you check this information?

Observation (Unit propagation). Consider the following examples

1. $(p \supset q \supset r) \wedge p \wedge \neg q$
2. $(p \wedge q) \vee \neg p \supset r$
3. $(p \wedge r) \vee (\neg q \wedge p) \vee (\neg r \wedge \neg p)$

Execute DPLL first without and then with your modification. Then compute how much iterations you saved

Pure literal – enhancement 2

Observation (Pure literal). Assume that a literal occurs only positively or only negatively. How would you modify the algorithm produced in the previous step to take into account this situation. When do you check this information?

Example (Pure literal). Consider the following examples

1. $(p \supset q \supset r) \land p \land q$
2. $((p \land q) \supset r) \land (p \supset r)$
3. $(p \land \neg r) \lor (q \land p) \lor (\neg r \land q)$

Execute DPLL first without and then with your modification. Then compute how many iterations you saved

Literal counting – enhancement 3

Observation (Literal counting). Assume that you count the number of time each single literal occurs in a formula. How would you modify the algorithm produced in the previous step to take into account this additional information? When do you compute this information? Is it guaranteed to improve performance?

Exercise (Literal counting). Consider the following examples

1. $(p \supset q \supset r) \land p \land \neg q$
2. $(p \land q) \lor \neg p \supset r$
3. $(p \land r) \lor (\neg q \land p) \lor (\neg r \land \neg p)$

Execute DPLL first without and then with your modification. Then compute how much iterations you saved

DPLL decision procedure – final

DPLL(φ, I)

while {L} ∈ φ

do φ ← **DPLL**(unit-propagate (φ|$_L$, I ∪ {L}) **end**;

while pure(L) **and** {L} ∈ φ

do φ ← **DPLL**(pure-literal-assign (φ|$_L$, I ∪ {L}) **end**;

if {} ∈ φ **then exit** {} **end**;

if φ = {} **then exit-return** I **end**;

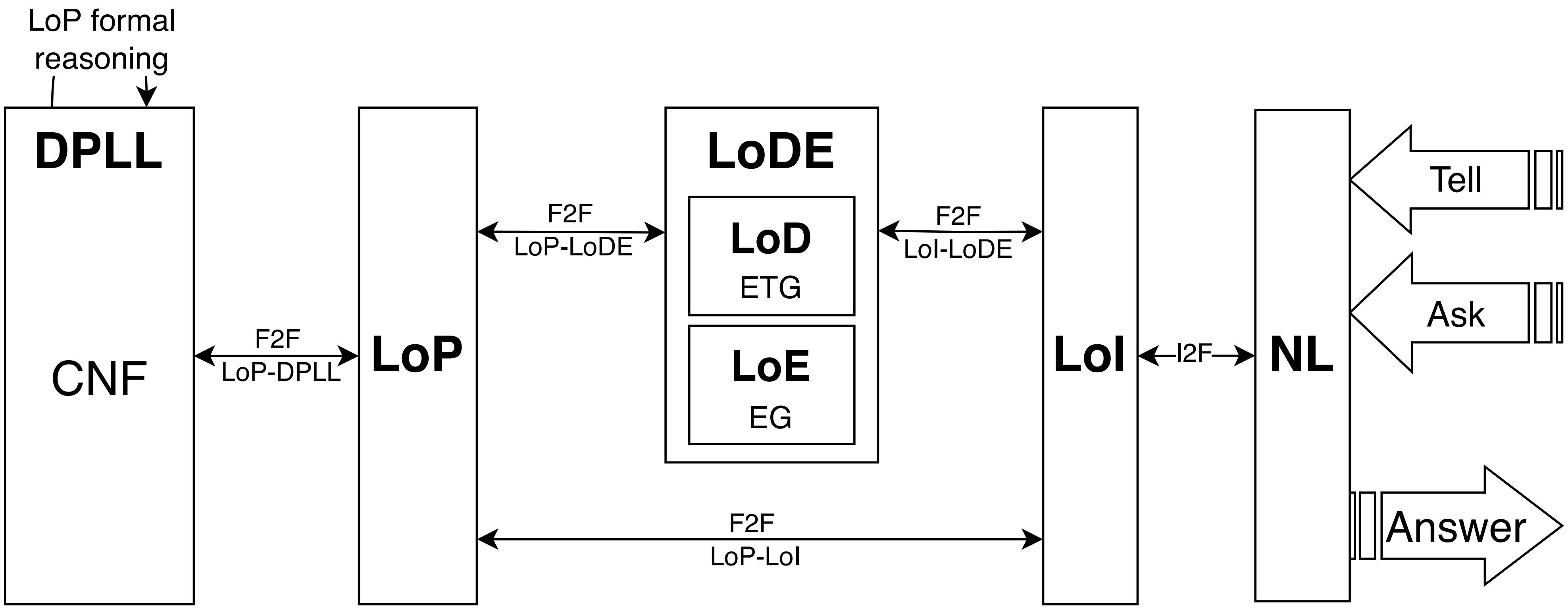
L ← select-literal(φ);

DPLL(φ|$_L$, I ∪ {L} or **DPLL**(φ|$_{\neg L}$, I ∪ {¬L})

Reasoning about propositions

- **Intuition**
- NL to LoI
- ALC / LoDE to LoI
- LoI to LoP
- LoP to CNF
- CNF reasoning
- **Reasoning about propositions – end-to-end**

Reasoning about propositions – the big picture



Notation. NL: Natural (informal) Language. I2F: Informal to Formal. F2F: Formal to Formal

Reasoning with propositions – exercise

Exercise (Reasoning with propositions). Ground the following sentence according to D and then apply DPLL to the grounded formula

$$\forall x.((King(x) \land Greedy(x)) \supset Evil(x))$$

with domain

$$D = \{John, Richard\}$$

UNIVERSITY OF TRENTO - Italy

Know dive

Reasoning with propositions – exercise

Exercise(Reasoning with propositions). Formalize the following sentence into FOL language, then ground it according to the domain D, assuming no synonyms, and apply DPLL to the grounded formula.

"Everyone who loves all animals is loved by someone"

$$D = \{John, Pauline, Simba, Sid\}$$

where

- Everyone, someone are variables ranging over the domain D_1 = {John, Pauline}
- Animal is a variable ranging over the domain D_2 = {Simba, Sid}

LoP- Reasoning

(T2MP)