Logic(s) for computer science - Week 13
Normal forms for First Order Logic - Part II

In the previous lecture we studied the Prenex normal form from First Order Logic. In this lecture we will see other normal forms for First Order Logic: conjunctive normal form, Skolem normal form and Clausal Skolem normal form.

Finally, we will see how to apply the binary resolution to check if a formula in FOLis satisfiable.

1 Closed formulae

Definition 1.1. *A formula $\varphi \in$ FOL is* closed *if free$(\varphi) = \emptyset$.*

In other words, if a formula does not have free variables, it is called closed. Closed formulae are also called *sentences*.

Definition 1.2. *A formula that is not closed is called* opened.

Example 1.1. *The formula $\forall x.P(x,x) \wedge \exists y.P(y,x)$ is a closed formula because free$(\forall x.P(x,x) \wedge \exists y.P(y,x)) = \emptyset$.*

The formula $\forall z.\exists y.(\neg(P(z,z) \wedge \neg P(z,y)) \wedge P(x,x))$ is not closed (is opened) because free$(\forall z.\exists y.(\neg(P(z,z) \wedge \neg P(z,y)) \wedge P(x,x))) = \{x\}$.

Definition 1.3. *Let $\varphi \in$ FOL be a formula and free$(\varphi) = \{x_1, \ldots, x_n\}$ the set of its free variables.*

The formula

$$\exists x_1.\exists x_2. \ldots . \exists x_n.\varphi$$

is is called the existential closure of the formula φ.

Remark 1.1. *The existential closure of a formula is a closed formula.*

Example 1.2. *The existential closure of the formula $\forall z.\exists y.(\neg(P(z,z) \wedge \neg P(z,y)) \wedge P(x,x))$ is $\exists x.\forall z.\exists y.(\neg(P(z,z) \wedge \neg P(z,y)) \wedge P(x,x))$.*

Definition 1.4. *Two formulae $\varphi_1 \in$ FOL and $\varphi_2 \in$ FOL are* equisatisfiable*, if:*

1. *either both φ_1 and φ_2 are satisfiable;*
2. *or neither φ_1 or φ_2 are satisfiable.*

In other qirds, the only cases in which the two formulae are not equisatisfiable are when one of the formula is satisfiable and the other not.

Theorem 1.1. *Any formula is equisatisfiable with its existential closure.*

Definition 1.5. *Let $\varphi \in$ FOL be a formula and free$(\varphi) = \{x_1, \ldots, x_n\}$ the set of its free variables.*

The formula

$$\forall x_1.\forall x_2.\ldots.\forall x_n.\varphi$$

is called universal closure *of the formula φ.*

Remark 1.2. *The universal closure of a formula is a closed formula.*

Example 1.3. *The universal closure of the formula $\forall z.\exists y.(\neg(P(z,z)\wedge\neg P(z,y))\wedge P(x,x))$ is $\forall x.\forall z.\exists y.(\neg(P(z,z)\wedge\neg P(z,y))\wedge P(x,x))$.*

Theorem 1.2. *A formula is valid if and only if its universal closure is valid.*

2 Skolem normal form

Definition 2.1 (FNS)**.** *A formula φ* is in Skolem normal form *(for short, SNF) if*

$$\varphi = \forall x_1.\ldots.\forall x_n.\varphi',$$

where:

1. *φ' does not contain quantifiers and*
2. *free$(\varphi') \subseteq \{x_1, \ldots, x_n\}$.*

In other words, a formula is in Skolem normal form if it contains only universal quantifiers found "in front" of the formula, and all variable occurrences in the formula are bounded (we have quantifiers for all variables that appear in the formula).

Remark 2.1. *A formula in SNF is closed because all free variables from the formula φ' are universal quantified in φ (because of condition 2), and therefore there are no free variables in φ.*

Example 2.1. *In the following we work on the signature $\Sigma = (\{P, Q, R\}, \{f, i, e\})$, where P, Q, R are predicate symbols of arities $2, 1$ and 3, f and i are functional symbols of arities 2 and respectively 1, and e is a functional symbol of arity 0 (constant).*

Examples of formulae in SNF:

$$\forall x.P(x, i(e)) \qquad \forall x.\forall y.\Big(P(f(x,e),y) \wedge \neg(R(x, i(f(y,y)), e) \vee Q(y))\Big)$$

Examples of formulae that are not in SNF:

$$\exists x.P(x,x) \qquad \forall x.\Big(Q(e) \wedge \neg(Q(x) \vee Q(y))\Big) \qquad Q(e) \wedge \forall x.Q(x)$$

In the case of the formulae that are not in SNF, the reasons why this holds are the following: the first formula contains existential quantifiers; the second formula contains a free occurence of variable y; and the third formula, the quantifier $\forall x$ is not at the "begining" of the formula.

Theorem 2.1 (Theorem of transformation in SNF)**.** *For any formula $\varphi \in$ `FOL`, there is a formula $\varphi' \in$ `FOL` such that:*

1. *φ' is in Skolem normal form;*
2. *φ and φ' are equisatisfiable.*

Proof sketch.
1. Compute the formula φ_1, in PNF equivalent to φ (using the theorem to transform a formula in PNF from the previous lecture);
2. Compute a formula φ_2, the existential closure of φ_1 (φ_2 which is equisatisfiable with φ_1 and therefore also with φ);
3. Apply the Skolemisation lemma on formula φ_2 (several times).

The result is a formula in SNF, equisatisfiable with the starting formula.

□

Lemma 2.1 (Skolem)**.** *Let $\varphi = \forall x_1.\forall x_2.\ldots.\forall x_k.\exists x.\varphi'$, where $k \geq 0$, $\varphi' \in$ `FOL` (φ' may contain other quantifiers). In other words, there are k universal quantifiers before the first existential quantifier.*

Let $f \in \mathcal{F}_k$ be a functional symbol of arity k that does not appear in φ (a fresh functional symbol).

We have that φ is equisatisfiable with

$$\forall x_1.\ldots.\forall x_k.(\sigma^\flat(\varphi')),$$

where $\sigma = \{x \mapsto f(x_1,\ldots,x_k)\}$.

Proof sketch. Considering that the formula φ is over the signature $\Sigma = (\mathcal{P},\mathcal{F})$, we observe that the newly obtained formula is over the signature $\Sigma' = (\mathcal{P},\mathcal{F} \cup \{f\})$ that contains, besides the predicate and functional symbols from Σ, the functional symbol f of arity k.

Direct implication:

We suppose that there is a Σ-structure S and an assignment α such that $S,\alpha \models \varphi$.

We find a Σ'-structure S' and an assignment α' such that $S',\alpha' \models \forall x_1.\forall x_2.\ldots.\forall x_k.(\sigma^\flat(\varphi'))$.

Attention! The structure S' is over the signature Σ' which is richer then the the signature of the structure S (it contains the new functional symbol f of arity k). On the other hand, the assignment α'may be the same as α.

Inverse implication:

We suppose there is an Σ'-structure S' and an assignment α' such that $S', \alpha' \models \forall x_1.\forall x_2.\ldots\forall x_k.(\sigma^\flat(\varphi'))$.

We find a Σ-structure S and assignment σ such that $S, \alpha \models \varphi.$. Again, the assignment σ may be the same as σ'.

□

Exercise 2.1. *Fill the technical details in the above proof.*

Example 2.2. *We compute a Skolem normal form for the formula*

$$\varphi = \forall x.\exists y.\forall z.\exists z'.(P(x,y) \leftrightarrow P(z,z')).$$

By the Lemma 2.1, we have that φ is equisatisfiable with

$$\varphi_1 = \forall x.\forall z.\exists z'.(P(x,g(x)) \leftrightarrow P(z,z')),$$

where g is a new functional symbol of arity 1.

Applying again the Lemma 2.1, we have that the formula φ_1 is equisatisfiable with

$$\varphi_2 = \forall x.\forall z.(P(x,g(x)) \leftrightarrow P(z,h(x,z))),$$

where h is a functional symbol of arity 2.

In conclusion, φ_2 is in SNF and is equisatisfiable with φ, so it is a Skolem normal form for φ.

Remark 2.2. *The signature of the Skolem normal form may be richer then the signature of the initial formula because of the addition of the Skolem symbols.*

3 Conjunctive normal form

Definition 3.1 (Literal)**.** *A formula $\varphi \in$ `FOL` is called* literal *if there is a predicate symbol $P \in \mathcal{P}_n$ if arity $n \geq 0$ and n the terms $t_1, \ldots, t_n \in \mathcal{T}$ such that*

$$\varphi = P(t_1, \ldots, t_n) \text{ sau } \varphi = \neg P(t_1, \ldots, t_n).$$

In other words, a literal is an atomic formula or its negation.

Example 3.1. *Examples of literals:*

$$P(x,x) \qquad \neg P(x,i(y)) \qquad \neg R(a,b,c) \qquad \neg P(x,x) \qquad Q(i(x))$$

$$R(a,f(x),b)$$

Examples of formulas that are not literals:

$$P(x,y) \wedge P(x,y) \qquad \neg\neg P(x,x) \qquad \forall x.P(x,x)$$

Definition 3.2 (Clause)**.** *A formula $\varphi \in$ FOL is called a* clause *if there are n literals $\varphi_1, \ldots, \varphi_n \in$ FOL such that*

$$\varphi = \varphi_1 \vee \varphi_2 \vee \ldots \vee \varphi_n.$$

Example 3.2.

$$P(x,x) \vee R(x,e,y) \vee \neg P(x, f(e,y)) \qquad P(x,x) \qquad \square \qquad P(x,x)$$

$$P(x,x) \vee R(x, f(e,e), y) \vee \neg P(x, f(e,y)) \vee \neg P(e, i(x))$$

Remark 3.1. *A particular case is the empty clause, denoted by $\square$, which is the disjunction of 0 literals. The empty clause is an unsatisfiable formula.*

Another particular case is represented by the literals. Any literal is a clause, being a conjunction of a single literal.

Definition 3.3 (CNF)**.** *A formula φ is* in clausal normal form *(or* in conjunctive normal form*) if there are $n \geq 1$ clauses $\varphi_1, \ldots, \varphi_n$ such that*

$$\varphi = \varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_n.$$

Example 3.3. *The following formulae are in CNF:*

$$(P(x,x) \vee Q(x)) \wedge (\neg P(x,y) \vee R(x,y,e))$$

$$(P(x,y) \vee Q(i(x)) \vee \neg Q(e)) \wedge (\neg P(x,x)) \wedge (\neg Q(f(z,z)) \vee R(x,z))$$

4 Clausal Skolem normal form

Definition 4.1. *A formula φ* is in clausal Skolem normal form *(for short CSNF) if*

1. *φ is in Skolem normal form and*
2. *φ' is in clausal normal form, where: $\varphi = \forall x_1. \ldots . \forall x_n . \varphi'$, and φ' does not have quantifiers (in other words, φ' is the subformula obtained from φ by removing the quantifiers).*

Example 4.1. *Examples of formulas in CSNF:*

$$\forall x. \forall y. \Big((P(x,x) \vee \neg Q(i(x)) \wedge (P(e,y) \vee \neg Q(e)) \Big)$$

$$\forall x. \forall y. \Big(Q(e) \wedge (\neg R(x,e,y) \vee Q(i(y))) \Big)$$

$$\forall x. \forall y. \forall z. (P(x,y) \wedge (Q(x) \vee R(x,y,z)) \wedge \neg Q(x)).$$

Examples of formulas that are not in CSNF:

$$\exists x.Q(x) \qquad \forall x.\Big(Q(e) \wedge (\neg R(x,y,z) \vee Q(y))\Big)$$

$$\forall x.\forall y.\Big(Q(e) \wedge \neg(Q(x) \vee Q(y))\Big)$$

Theorem 4.1. *For any formula $\varphi \in$ FOL in SNF there is a formula $\varphi' \in$ FOL such that:*

1. *φ' is in CSNF and*
2. *$\varphi \equiv \varphi'$.*

Proof sketch. We apply from left to right the following equivalences:

1. $\varphi_1 \leftrightarrow \varphi_2 \equiv (\varphi_1 \rightarrow \varphi_2) \wedge (\varphi_2 \rightarrow \varphi_1)$;
2. $\varphi_1 \rightarrow \varphi_2 \equiv \neg\varphi_1 \vee \varphi_2$;
3. $\neg\neg\varphi \equiv \varphi$;
4. $\neg(\varphi_1 \vee \varphi_2) \equiv \neg\varphi_1 \wedge \neg\varphi_2$;
5. $\neg(\varphi_1 \wedge \varphi_2) \equiv \neg\varphi_1 \vee \neg\varphi_2$;
6. $\varphi_1 \vee (\varphi_2 \wedge \varphi_3) \equiv (\varphi_1 \vee \varphi_2) \wedge (\varphi_1 \vee \varphi_3)$.

Also, we can use the associativity and commutativity of the connectors $\vee$ and $\wedge$.

The first two equivalences remove the quantifiers $\leftrightarrow$ and $\rightarrow$.

The following three equivalences ensure that the negations are only over the atomic formulae.

The last equivalence ensures that the $\vee$ does not appear over $\wedge$.

Finally, we obtain a formula in which at the root (besides the universal operators) we have $\wedge$, followed by a layer of $\vee$, followed by $\neg$, followed by the atomic formulae, which means that the formula is in CSNF.

□

Remark 4.1. *A consequence of the theorems 2.1 and 4.1 is the fact that for any formula $\varphi \in$ FOL there is a formula $\varphi' \in$ FOL such that φ' is in $CSNF$ and the formulae φ and φ' are equisatisfiabile.*

In order to obtain such a formula, we have to follow the following steps:

1. *Compute the formula φ_1 in PNF equivalent with φ;*
2. *Compute φ_2 the existential closure forφ_1;*
3. *Apply the Skolemisation lemma in order to remove the existential quantifiers and we obtain the formula φ_3;*
4. *Apply the equivalences from the theorem 4.1 in order to put the formula φ_3 in CSNF.*

5 An example of transforming a formula in CSNF

We are interested in establishing the validity of the formula

$$\varphi = \Big(\forall x.\big(Q(x) \leftrightarrow \neg Q(i(x))\big)\Big) \rightarrow \Big(\forall x.\big(Q(x) \rightarrow Q(i(i(x)))\big)\Big).$$

It is easy to see that a formula is valid if and only if its negation is not satisfiable. Therefore, in order to establish that φ is valid, it is enough the show that

$$\neg\varphi = \neg\Bigg(\Big(\forall x.\big(Q(x) \leftrightarrow \neg Q(i(x))\big)\Big) \rightarrow \Big(\forall x.\big(Q(x) \rightarrow Q(i(i(x)))\big)\Big)\Bigg)$$

is not satisfiable.

We will compute a CSNF of the formulai $\neg\varphi$, CSNF about which we know that it is equisatisfiable with the formula $\neg\varphi$. The first step is to find a PNF:

$$\begin{aligned}
\neg\varphi = \quad & \neg\Bigg(\Big(\forall x.\big(Q(x) \leftrightarrow \neg Q(i(x))\big)\Big) \rightarrow \Big(\forall x.\big(Q(x) \rightarrow Q(i(i(x)))\big)\Big)\Bigg) \\
\equiv \quad & \neg\Bigg(\neg\Big(\forall x.\big(Q(x) \leftrightarrow \neg Q(i(x))\big)\Big) \vee \Big(\forall x.\big(Q(x) \rightarrow Q(i(i(x)))\big)\Big)\Bigg) \\
\equiv \quad & \Big(\neg\neg\forall x.\big(Q(x) \leftrightarrow \neg Q(i(x))\big)\Big) \wedge \Big(\neg\forall x.\big(Q(x) \rightarrow Q(i(i(x)))\big)\Big) \\
\equiv \quad & \Big(\forall x.\big(Q(x) \leftrightarrow \neg Q(i(x))\big)\Big) \wedge \Big(\exists x.\neg\big(Q(x) \rightarrow Q(i(i(x)))\big)\Big) \\
\equiv \quad & \forall x.\Big(\big(Q(x) \leftrightarrow \neg Q(i(x))\big) \wedge \exists x.\neg\big(Q(x) \rightarrow Q(i(i(x)))\big)\Big) \\
\equiv \quad & \forall x.\Big(\big(Q(x) \leftrightarrow \neg Q(i(x))\big) \wedge \exists x'.\neg\big(Q(x') \rightarrow Q(i(i(x')))\big)\Big) \\
\equiv \quad & \forall x.\exists x'.\Big(\big(Q(x) \leftrightarrow \neg Q(i(x))\big) \wedge \neg\big(Q(x') \rightarrow Q(i(i(x')))\big)\Big) \\
\equiv \quad & \forall x.\exists x'.\Big(\big(Q(x) \rightarrow \neg Q(i(x))\big) \wedge \big(\neg Q(i(x)) \rightarrow Q(x)\big) \wedge \neg\big(Q(x') \rightarrow Q(i(i(x')))\big)\Big) \\
\equiv \quad & \forall x.\exists x'.\Big(\big(\neg Q(x) \vee \neg Q(i(x))\big) \wedge \big(\neg\neg Q(i(x)) \vee Q(x)\big) \wedge \neg\big(\neg Q(x') \vee Q(i(i(x')))\big)\Big)
\end{aligned}$$

Therefore, a prenex normal form for the formula $\neg\varphi$ is

$$\varphi_1 = \forall x.\exists x'.\Big(\big(\neg Q(x)\vee\neg Q(i(x))\big)\wedge\big(\neg\neg Q(i(x))\vee Q(x)\big)\wedge\neg\big(\neg Q(x')\vee Q(i(i(x')))\big)\Big).$$

We continue by computing a SNF of the formula φ_1, by applying the Skolemization lemma, and using a fresh Skolem symbol g, of arity 1:

$$\varphi_1 = \forall x.\exists x'.\Big(\big(\neg Q(x)\vee\neg Q(i(x))\big)\wedge\big(\neg\neg Q(i(x))\vee Q(x)\big)\wedge\neg\big(\neg Q(x')\vee Q(i(i(x')))\big)\Big)$$

equisatisfiable with

$$\forall x.\Big(\big(\neg Q(x) \vee \neg Q(i(x))\big) \wedge \big(\neg\neg Q(i(x)) \vee Q(x)\big) \wedge \neg\big(\neg Q(g(x)) \vee Q(i(i(g(x))))\big)\Big)$$

Therefore, a SNF of the formula $\neg\varphi$ is the formula $\varphi_2 = \forall x.\Big(\big(\neg Q(x) \vee \neg Q(i(x))\big) \wedge \big(\neg\neg Q(i(x)) \vee Q(x)\big) \wedge \neg\big(\neg Q(g(x)) \vee Q(i(i(g(x))))\big)\Big)$. We continue, to find a CSNF for the formula φ_2:

$$\begin{aligned}
\varphi_2 = \quad & \forall x.\Big(\big(\neg Q(x) \vee \neg Q(i(x))\big) \wedge \big(\neg\neg Q(i(x)) \vee Q(x)\big) \wedge \neg\big(\neg Q(g(x)) \vee Q(i(i(g(x))))\big)\Big) \\
\equiv \quad & \forall x.\Big(\big(\neg Q(x) \vee \neg Q(i(x))\big) \wedge \big(\neg\neg Q(i(x)) \vee Q(x)\big) \wedge \big(\neg\neg Q(g(x)) \wedge \neg Q(i(i(g(x))))\big)\Big). \\
\equiv \quad & \forall x.\Big(\big(\neg Q(x) \vee \neg Q(i(x))\big) \wedge \big(Q(i(x)) \vee Q(x)\big) \wedge \big(Q(g(x)) \wedge \neg Q(i(i(g(x))))\big)\Big). \\
\equiv \quad & \forall x.\Big(\big(\neg Q(x) \vee \neg Q(i(x))\big) \wedge \big(Q(i(x)) \vee Q(x)\big) \wedge \big(Q(g(x))\big) \wedge \big(\neg Q(i(i(g(x))))\big)\Big).
\end{aligned}$$

6 Binary resolution

In order to test the satisfiability of a formula in CSNF, we can use the deductive system described in this section.

Definition 6.1. *a term $t \in \mathcal{T}$ is called* ground term *if $vars(t) = \emptyset$.*

Definition 6.2. *A substitution $\sigma = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$ is called* ground substitution *if $t_1, \ldots, t_n$ are ground terms.*

Definition 6.3. *The binaru resolution is a deductive system*[1] *for clauses, with the following inference rule, called binary resolution :*

$$\frac{C_1 \vee P(t_1, \ldots, t_n) \qquad C_2 \vee \neg P(t'_1, \ldots, t'_n)}{\sigma_1^\flat(C_1) \vee \sigma_2^\flat(C_2)} \quad \begin{array}{l} \sigma_1 \text{ E SUBST. DE BAZĂ} \\ \sigma_2 \text{ E SUBST. DE BAZĂ} \\ \sigma_1^\sharp(t_i) = \sigma_2^\sharp(t'_i) \text{ PENTRU ORICE } 1 \leq i \leq n \end{array}$$

Theorem 6.1 (Theorem of binary resolution)**.** *A formula φ in CSNF is unsatisfiable if and only if we can obtain $\square$ by binary resolution starting from the clauses of φ.*

Example 6.1. *We continue the example from Section 5 in order to show that the formula φ is valid. In the previous section we computed a CSNF $\varphi_3 = \forall x.\Big(\big(\neg Q(x) \vee \neg Q(i(x))\big) \wedge \big(Q(i(x)) \vee Q(x)\big) \wedge \big(Q(g(x))\big) \wedge \big(\neg Q(i(i(g(x))))\big)\Big)$ for the formula $\neg\varphi$.*

We will show, using the binary resolution, that the formula φ_3 is unsatisfiable:

1. $\neg Q(x) \vee \neg Q(i(x))$;
2. $Q(i(x)) \vee Q(x)$;
3. $Q(g(x))$;

[1] In fact, the deductive system also contains another rule, called factorization, that we will see later

4. $\neg Q(i(i(g(x))))$;

5. $\neg Q(i(g(e)))$ *(3,1,*$\sigma_1 = \{x \mapsto e\}$*,*$\sigma_2 = \{x \mapsto g(e)\}$*,* $\sigma_1^\flat(Q(g(x))) = \sigma_2^\flat(Q(x))$*);*

6. $Q(i(i(g(e))))$ *(2,5,*$\sigma_1 = \{x \mapsto i(g(e))\}$*,* $\sigma_2 = \{\}$*,* $\sigma_1^\flat(Q(x)) = \sigma_2^\flat(Q(i(g(e)))))$*;*

7. $\square$ *(6, 4,* $\sigma_1 = \{\}$*,* $\sigma_2 = \{x \mapsto e\}$*,* $\sigma_1^\flat(Q(i(i(g(e))))) = \sigma_2^\flat(Q(i(i(g(x))))))$*.*

Because we arrived at the empty clause, by Theorem 6.1 that the formula $\forall x.\Big(\big(\neg Q(x) \vee \neg Q(i(x))\big) \wedge \big(Q(i(x)) \vee Q(x)\big) \wedge \big(Q(g(x))\big) \wedge \big(\neg Q(i(i(g(x))))\big)\Big)$ *is unsatisfiable.*

But the above formula is a CSNF for $\neg\varphi$*, and therefore the two formulae are equisatisfiable. Therefore* $\neg\varphi$ *is also unsatisfiable. Therefore the start formula,*

$$\varphi = \Big(\forall x.\big(Q(x) \leftrightarrow \neg Q(i(x))\big)\Big) \rightarrow \Big(\forall x.\big(Q(x) \rightarrow Q(i(i(x)))\big)\Big),$$

is valid.