LOGIC IN COMPUTER SCIENCE. (Fall 2016)

⑤ Is this SAT?

$$\forall x \quad \neg p(x,x) \;\wedge$$
$$\forall x \forall y \forall z \quad (p(x,y) \wedge p(y,z) \rightarrow p(x,z)) \;\wedge$$
$$\forall x \exists y \quad p(x,y) \;\wedge$$
$$\forall x \forall y \quad (p(x,y) \longrightarrow \exists z \; (p(x,z) \wedge p(z,y)))$$

Yes $p(x,y) = 1 \Leftrightarrow x < y$

& Domain $= \mathbb{Q}$

⑥ a) How to prove that $F \not\models G$ (First order formulas)

Giving a counterexample of an interpretation I such that $I \models F$ but $I \not\models G$.

b) Same for $F \models G$

We know that $F \models G \Leftrightarrow F \wedge \neg G$ unsat, to prove it we turn $F \wedge \neg G$ into a clausal ~~form~~ form and obtain the empty clause from it under resolution and factoring.

c) F is $\forall x\, p(a,x) \wedge \exists y \neg q(y)$
G is $\exists v \exists w \; (\neg q(w) \wedge p(v,a))$ } $F \models G$?

[F] $\forall x\, p(a,x) \wedge \exists y \neg q(y)$

$p(a,x) \wedge \neg q(c_y)$

① $p(a,x)$

② $\neg q(c_y)$

[¬G] $\neg(\exists v \exists w \; (\neg q(w) \wedge p(v,a)))$

$\forall v \forall w \; (q(w) \vee \neg p(v,a))$

$q(w) \vee \neg p(v,a)$

$$\frac{\neg q(c_y) \qquad q(w) \vee p(v,a)}{}$$

$$mgu = \{c_y = w\}$$

$$\frac{p(v,a) \qquad \neg p(a,x)}{}$$

$$mgu = \begin{cases} v = a \\ x = a \end{cases}$$

$\square$

$F \models G$ ↗

d) F is $\forall x \exists y\ p(x,y)$
G is $\exists y \forall x\ p(x,y)$
} $F \models G$?

F | $p(x, C_y)$ ¬G | $\forall y \exists x \neg p(x,y)$

$\neg p(z, C_z)$

$p(x, C_y)$ $\quad$ $\neg p(z, C_z)$

we can't obtain the empty clause.

• Counter example :

$p(x,y) = 1 \iff x == y$ $\quad$ Domain $= \{a, b\}$

x	y	p(x,y)
a	a	1
a	b	0
b	b	1
b	a	0

$I \not\models F$ but $I \not\models G$.

(7) return (L , A) minimal number of coins in L to reach amount A.

```
return ( L , A) :-
    length ( L, sizeL),
    length ( Vars, sizeL),
    Vars ins 0..A,
    scalar_product ( Vars , L , #= , A),
    sum_list ( Vars, Sum),
    labeling ( [min(Sum)], Vars ),
    write (Sum) , nl , !.
```

④ brother(joan, pere).
father(enric, joan).
uncle(N,U) :- father(N,F), brother(F,U).

Express as a set of first order clauses P.

$\forall N \forall U \; uncle(N,U) \leftarrow \exists F (father(N,F) \wedge brother(F,U)) \equiv$

$\forall N \forall U \; uncle(N,U) \leftarrow \neg \exists F (father(N,F) \wedge brother(F,U)) \equiv$

$P \equiv (\forall N \forall U \; uncle(N,U)) \vee \forall F (\neg father(N,F) \vee \neg brother(F,U))$

- Prove that $\exists x \exists y \; uncle(x,y)$ is a logical consequence of P.

$P \models \exists x \exists y \; uncle(x,y) \Leftrightarrow P \wedge \neg(\exists x \exists y \; uncle(x,y))$ unsat

$\Leftrightarrow \square \in P \wedge \neg(\exists x \exists y \; uncle(x,y))$

① brother(joan, pere)
② father(enric, joan)
③ $uncle(N,U) \vee \neg father(N,F) \vee \neg brother(F,U)$

④ $\neg(\exists x \exists y \; uncle(x,y))$
↳ $\forall x \forall y \; uncle(x,y)$
↳ $\neg uncle(x,y)$

$uncle(N,U) \vee \overline{father}(N,F) \vee \overline{brother}(F,U)$ — $\overline{uncle}(x,y)$

mgu $\{N=x, U=y\}$ → $\overline{father}(N,F) \vee \overline{brother}(F,y)$ — brother(joan, pere)

mgu $\{F=joan, y=pere\}$ → $\overline{father}(x,joan)$ — father(enric, joan)

mgu $\{x=enric, F=joan\}$ → $\square$

$x = enric$

$y = pere$

cláusula de Horn con ningún literal positivo

⑤ True or false?

A) $\forall x \exists y (p(x, \underbrace{f(y)}_{x}) \wedge \underbrace{\neg p(x,y)}_{1})$ is sat.

$\underbrace{\quad\quad\quad\quad\quad\quad\quad\quad}_{1}$

(f)

val	output
a	b
b	a

(p)

val 1	val 2	output
a	a	1
a	b	0
b	a	0
b	b	1

True

$Domain_I = \{a, b\}$

Alternativa:

True for $D_I = \{a, b\}$

$p_I(a,a) = 1$
$p_I(a,b) = 0$
$p_I(b,a) = 1$
$p_I(b,b) = 0$

$f_I(a) = a$
$f_I(b) = a$

B) $\underbrace{\forall x \forall y \exists z \; q(x,z,y)}_{F} \models \underbrace{\forall x \exists z \forall y \; q(x,z,y)}_{G}$

$F \models G$? False

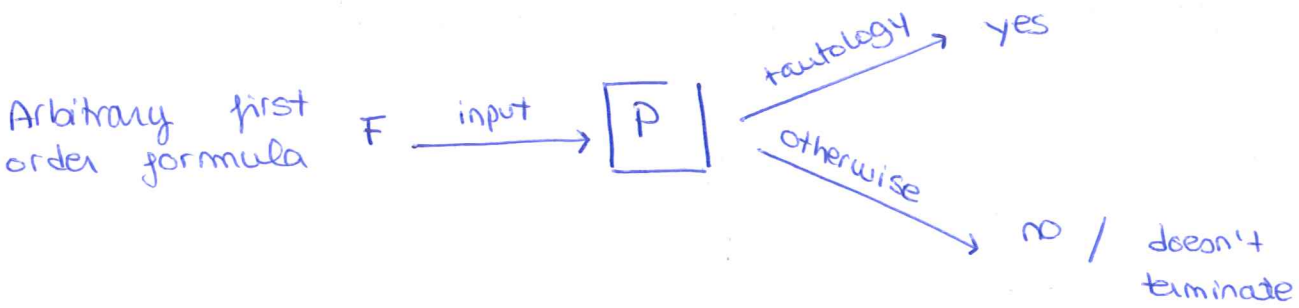
- Counterexample:

$D_I = \{a, b\}$ $\quad q(x,z,y) = 1 \Leftrightarrow y = z$

$I \models \forall x \forall y \exists z \; q(x,z,y)$ however,

$I \not\models \forall x \exists z \forall y \; q(x,z,y)$

(6)



Is this possible?

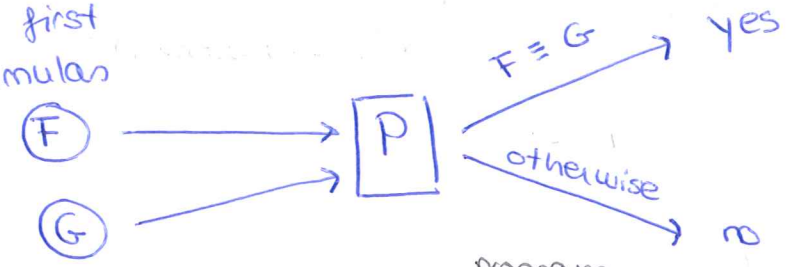
Yes, because we have that F tautology $\Leftrightarrow \neg F$ unsat $\Leftrightarrow$ $\Leftrightarrow$ $S = \text{clausal} - \text{form}(\neg F)$ unsat $\Leftrightarrow$ the empty clause is in the closure under resolution and factoring of S.

P behaviour:

- we find the empty clause $\Rightarrow$ output : yes
- There isn't an empty clause and the closure under resolution and factoring of S is finit $\Rightarrow$ output : no
- otherwise $\Rightarrow$ it does not terminate.

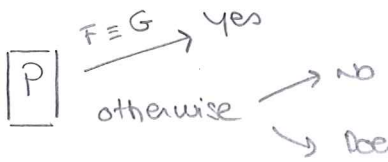
LOGIC IN COMPUTER SCIENCE (Spring 2016)

(4) Arbitrary first order formulas



How would you implement this ~~problem~~ program?

That program does not exist because this problem is undecidable, it is only semi-decidable: we can get:



How to solve it:

① Convert $(F \wedge \neg G) \vee (\neg F \wedge G)$ into its clausal form S_0 we have $F \equiv G \Leftrightarrow S_0$ unsat.

② Compute the closure under resolution and factoring of S_0 by levels, in successive steps for $i = 0, 1, 2 \ldots$

a) $\square \in S_i \Rightarrow F \equiv G \Rightarrow$ output yes

b) Otherwise, obtain S_{i+1} by adding all clauses obtained by one step of resolution or factoring on clauses in S_i

c) If no new clause was obtained from $S_i \Rightarrow$ no
else $\Rightarrow$ go to 2a with the next i.

(5) Formalize and prove by resolution that E is a logical consequence of the other four.

A) $logic(P) \rightarrow \overline{football(P)}$

B) $player(P) \wedge brother(P, P1) \rightarrow football(P1)$

C) $player(messi) \wedge brother(messi, ney)$

D) $logic(ney)$

E) $football(robert) \wedge logic(robert)$

A) $\overline{logic}(P) \vee \overline{football}(P)$

B) $\overline{player}(P) \vee \overline{brother}(P, P1) \vee football(P1)$

C1) $player(messi)$

C2) $brother(messi, ney)$

D) $logic(ney)$

- We want to prove that $A \wedge B \wedge C \wedge D \neq E$, then we have to prove that $A \wedge B \wedge C \wedge D$ is unsat.

By resolution:

(A) $\overline{logic}(P) \vee \overline{football}(P)$ — (D) $logic(ney)$

$mgu = \{P = ney\} \rightarrow \overline{football}(ney)$

(B) $\overline{player}(P) \vee \overline{brother}(P, P1) \vee football(P1)$ — (C1) $player(messi)$

$mgu = \{P = messi\} \rightarrow \overline{brother}(messi, P1) \vee football(P1)$ — $brother(messi, ney)$

$mgu = \{P1 = ney\} \rightarrow football(ney)$

$football(ney)$ — $\overline{football}(ney)$

$\square$ ← unsat

Proved by resolution

⑥ Graph coloring program.

makeConstraints recursively with #\= and nth1(I, L, X)

```
:- use_module(library(clpfd)).
numVertices(5).
edges([1-2, 1-3, 2-3, 2-4, 2-5, 3-5]).
numColors(3).

main :- numVertices(N), edges(Edges),
        numColors(K), length(Vars, N),
        Vars ins 1..K,
        makeConstraints(Edges, Vars),
        label(Vars),
        write(Vars), nl.

makeConstraints([X-Y|Edges], Vars):-
        nth1(X, Vars, ColorX),
        nth1(Y, Vars, ColorY),
        ColorX #\= ColorY,
        makeConstraints(Edges, Vars).
makeConstraints([], _):-!.
```

⑤ Is it decidable whether $I \models F$?

A) D_I is a finite set?

Yes, only a finite number of cases have to be checked to compute $eval_I(F)$

B) D_I is the integers and well-known operations + or * and predicates > and =.

NO This is undecidable in general for equations $P=0$ if P can be an arbitrary polynomial with products between variables such that $x^2y + 2z^5 + \ldots$. But, it would be decidable if one could evaluate arbitrary formulas in the integers!

We can express any polynomial such that

sum(prod(prod(x, prod(x,x)) ... such that

$P=0$ has no integer solution $\Leftrightarrow$ $I \models \forall x,y,z \ldots \neg eq_I(T_P, zero_I)$

where D_I are the integers and eq_I, $zero_I$, sum_I, $prod_I$ are the functions $=$, 0, $+$ and $*$.

C) F is a set of Horn clauses.

Being a set of Horn clauses does not help, the above example is a unit (Horn) clause

⑥ A) $F \not\models G$? By giving a counterexample such that $I \models F$ but $I \not\models G$.

B) $F \models G$? By proving that $F \wedge \neg G$ is unsat $\Leftrightarrow$ = S = clausal-form$(F \wedge \neg G)$ unsat $\Leftrightarrow$ We can obtain the empty clause from S by resolution and factoring.

És decidible per equacions quadràtiques amb una sola variable.

c) F is $\forall x\, p(a,x) \wedge \exists y\, \neg q(y)$
G is $\exists v \exists w\, \neg q(w) \wedge p(v,a)$
} $F \not\models G$?

- We want to prove $F \not\models G$: $F \not\models G \Leftrightarrow F \wedge \neg G$ unsat
- We turn it into clausal form:

(F) $p(a,X) \wedge \neg q(c_y)$ (¬G) $q(w) \vee \bar{p}(v,a)$

$p(a,X)$

$\neg q(c_y)$ ← Skolem constant

- If we obtain the empty clause by resolution and factoring of $F \wedge \neg G$, then it is unsat, then $F \models G$.

$q(w) \vee \bar{p}(v,a)$ $\quad$ $\neg q(c_y)$

$mgu = \{w = c_y\}$ → $\bar{p}(v,a)$ $\quad$ $p(a,x)$

$mgu = \{v = a, x = a\}$ → $\square$

Proved by resolution.

d) F is $\forall x \exists y\, p(x,y)$
G is $\exists y \forall x\, p(x,y)$
} $F \not\models G$?

No $\quad D_I = \{a, b\}$ $\quad p_I(a,a) = 1$
$b,b = 1$
otherwise 0

$I \models F$ but $I \not\models G$.

(4) Complete the following code in Prolog.

```
roomsOverlap(D, Hour, Room, Sched) :-
        member([_,_,D2,Hour2,Room], Sched),
        HourEnd is Hour + D - 1,
        HourEnd2 is Hour2 + D2 - 1,
        between(H, End, X),
        between(H2, HourEnd2, X).

blockingProblem(Hour, S, D) :-
        End is H+D-1,   member(J, S),
        blocking(J, X),  between(Hour, End, X).

attendantsOverlap(S, D, Hour, Sched) :-
        member([_, S2, D2, H2, _], Sched),
        member(J, S)  , member(J, S2),
        End is Hour + D - 1, End2 is H2 + D2 - 1,
        between(Hour, End, X) , between(H2, End2, X).

room(N, R) :- N < 26, !, numSmallRooms(S),
        numLargeRooms(L), T is S+L,
        between(1, T, R).

room(N, R) :- N < 51, numSmall(S), Large(L),
        T is S+L, K is S+1,
        between(K, T, R).
```

LOGIC IN COMPUTER SCIENCE (Spring 2015)

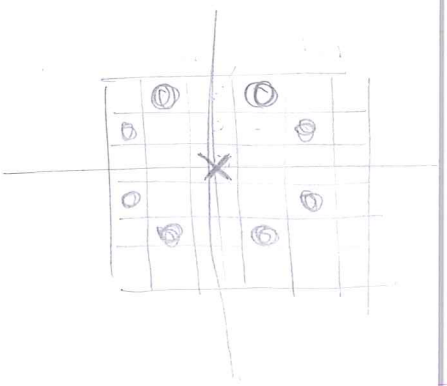
④ Prolog: shortest([I1, J1], [I2, J2]) $I, J, \in 1..3$

horse: [I1, J1] (from) — [I2, J2] (to)

```
shortest(From, To) :- path(From, To, [From], Path),
                      length(Path, N),
                      between(1, 64, N),
                      write(Path), nl.

path(C, C, Path, Path).
path(From, To, UntilNow, FinalPath) :-
    oneStep(From, Next),
    \+ member(Next, UntilNow),
    path(Next, To, [Next|UntilNow], FinalPath).

oneStep([I1-J1], [I2-J2]) :-
    member([A,B], [[1,2], [2,1]]),
    member(signA, [-1, 1]),
    member(signB, [-1, 1]),
    I2 is signA*A + I1,
    J2 is signB*B + J1
    between(1, 8, I2),
    between(1, 8, J2).
```



símbolos de predicado

símbolos de función

⑤a) $P = \{p^2, q^1\}$ $\mathcal{F} = \{f^2, g^1, a^0, b^0, c^0\}$

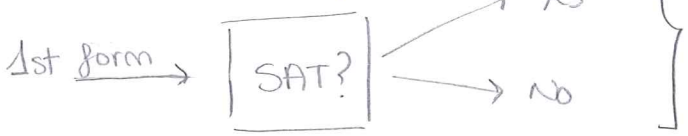
How many (≠) atoms without variables can be constructed using P and $\mathcal{F}$?

→ Todo símbolo de predicado de aridad cero

- Atoms → símbolo de predicado con aridad cero ⇒ no hay
  → $p(t_1 \dots t_n)$ $t_1 \dots t_n$ términos y p símbolo de predicado de aridad $n>0$.
- Términos → variables, constantes, $f(t_1 \dots t_n)$ $t_1 \dots t_n$ términos y f símbolo de función $n>0$.

infinitos porque podía ir anidando funciones.

⑤ b) Is the satisfiability of first-order formulas without variables decidable? Why?



Yes. If F has no variables, then let S be the clausal form of F, is finite and has no variables either.

We know that F is unsat ⇔ S unsat ⇔ □ ∈ ResFact(S). But since S has no variables, resolution here is just propositional resolution, and factoring is just eliminating repeated literals in a clause. Therefore, one can consider each different atom in S as a propositional predicate symbol, getting a propositional clause set prop(S) and □ ∈ ResFact(S) ⇔ □ ∈ PropositRes(prop(S)) which is decidable.

⑥ 1. All red horses run fast or there is a horse that does not run fast
2. There is at least one red horse.

a) Write first order formulas F_1 and F_2.

F_1 is $(\forall x\ red(x) \rightarrow fast(x)) \vee (\exists y\ \neg fast(y))$

F_2 is $\exists x\ red(x)$

b) F_1 tautology? F tautology ⇔ ¬F unsat ⇔ □ ∈ ResFact(¬F)

$\neg F_1$ is $\neg(\forall x\ red(x) \rightarrow fast(x)) \wedge \neg(\exists y\ \neg fast(y))$

$\equiv (\exists x\ red(x) \wedge \neg fast(x)) \wedge (\forall y\ fast(y))$

We have to find the empty clause in the closure under resolution and factoring.

$\Rightarrow$ $red(C_x)$, $\neg fast(C_x)$, $fast(y)$

(C_x: Skolem constants)

$$\frac{\neg fast(C_x),\quad fast(y)}{\square \qquad mgu = \{y = C_x\}}$$

⑥ c) Do we have $F_1 \not\models F_2$? Prove it formally.

$$\exists I \quad I \models F_1 \rightarrow I \not\models F_2$$

No, since we have proved that F_1 is a tautology, we only have to find an interpretation I such that $I \not\models F_2$

Counterexample:

$D_I = \{e\}$ $\quad red_I(e) = 0.$

LOGIC IN COMPUTER SCIENCE (Fall 2014)

⑥ F is $\exists x \forall y \exists z \; p(z,y) \wedge \neg p(x,y)$

a) Give a model I with $D_I = \{a, b, c\}$

(I)										
x	a	a	a	b	b	b	c	c	c	
y	a	b	c	b	a	c	c	a	b	
$p_I(x,y)$	0	0	0	1	1	1	1	1	1	

← The model is independent of this.

b) Is it true that $F \not\models \forall x \, p(x,x)$?

Nope, the model given ↑ of F does not satisfy $\forall x \, p(x,x)$

c) Is there any model of F with a single-element domain?

$D_I = \{a\}$ $\quad x = a, y = a, z = a$:

This is impossible because to satisfy F we would need $p(a,a) = 1$ and also $p(a,a) = 0$.

⑦ "The barber shaves all men that do not shave themselves, and only these men"

Formalize it in LFO and prove that the barber is a woman.

- shaves(x, y) : x shaves y
- man(x) : x is a man

} Definitions

* Formalization:

F: $\forall x \, (shaves(b,x) \leftrightarrow (man(x) \wedge \neg shaves(x,x)))$

where b is the barber.

* Prove that the barber is a woman

We have to prove $F \models \neg man(b) \Leftrightarrow F \wedge man(b)$ unsat

$\Leftrightarrow ResFact(F \wedge man(b))$ unsat $\Leftrightarrow \square \in ResFact(F \wedge man(b))$

① Get clausal forms:

$\forall x \, (shaves(b,x) \leftrightarrow (man(x) \wedge \neg shaves(x,x)))$

$\equiv \neg shaves(b,x) \vee (man(x) \wedge \neg shaves(x,x)) \wedge$

$shaves(b,x) \vee \neg man(x) \vee shaves(x,x)$

We get:

a) $\neg shaves(b,x) \vee man(x)$

b) $\neg shaves(b,x) \vee \neg shaves(x,x)$

c) $shaves(b,x) \vee \neg man(x) \vee shaves(x,x)$

And also:

d) $man(b)$

② We try to get the empty clause:

1) Res(d, c) $\sigma = mgu\{x = b\}$

$shaves(b,b) \vee shaves(b,b) \equiv shaves(b,b)$

2) Res(b, 1) $\sigma = mgu\{x = b\}$

$\neg shaves(b,b)$

3) Res(1, 2) $\longrightarrow$ we get the empty clause!

Proved by resolution.

LOGIC IN COMPUTER SCIENCE (Spring 2014)

(4) Prove in LPO that sentence 5 is a logical consequence of the first four.

1) All dogs make noise
2) Anyone who has a cat has no mice
3) Neurotic people do not have noisy animals
4) John has either a cat or a dog.
5) If John is neurotic he has no mice.

- Formalization.

1) $\forall x \quad dog(x) \rightarrow noisy(x)$
2) $\forall x \, (\exists y \; has(x,y) \wedge cat(y)) \rightarrow (\neg \exists z \; has(x,z) \wedge mouse(z))$
3) $\forall x \, (neurotic(x) \rightarrow \neg \exists y \; has(x,y) \wedge noisy(y))$
4) $\exists x \; has(John, x) \wedge (cat(x) \vee dog(x))$
5) $neurotic(John) \rightarrow \neg \exists x \; has(John,x) \wedge mouse(x)$

$\neg$5) $\neg(neurotic(John) \rightarrow \neg \exists x (has(John,x) \wedge mouse(x))) \equiv$
$\neg(\neg neurotic(John) \vee \neg \exists x (has(John,x) \wedge mouse(x))) \equiv$
$neurotic(John) \wedge (\exists x \; has(John,x) \wedge mouse(x))$

- Transforming into clausal.

1) $\neg dog(x) \vee noisy(x)$ ✓
2) $has(X, Y) \wedge cat(Y) \rightarrow \forall z \; \neg has(X,z) \vee \neg mouse(z)$
$\equiv \neg has(X, Y)) \vee \neg cat(Y)) \vee (\neg has(X,z) \vee \neg mouse(z))$ ✓
3) $\neg neurotic(x) \vee \neg has(x,y) \vee \neg noisy(y)$ ✓
4) a. $has(John, C_x)$ ✓
b. $cat(C_x) \vee dog(C_x)$ ✓

not 5) a. $neurotic(John)$ ✓
b. $has(John, C_z)$
c. $mouse(C_z)$

• Resolution.

A) Res(3, 5a) $\sigma = mgu\{X = John\}$

$\neg has(John, Y) \vee \neg noisy(Y)$

B) Res(A, 4a) $\sigma = mgu\{Y = C_x\}$

$\neg noisy(C_x)$

C) Res(B, 1) $\sigma = mgu\{X = C_x\}$

$\neg dog(C_x)$

D) Res(C, 4b) $\sigma = mgu\{C_x = C_x\}$

$cat(C_x)$

E) Res(D, 2) $\sigma = mgu\{Y = C_x\}$

$\neg has(X, C_x) \vee \neg has(X, Z) \vee \neg mouse(Z)$

F) Res(E, 4a) $\sigma = mgu\{X = John\}$

$\neg has(John, Z) \vee \neg mouse(Z)$

G) Res(F, 5b) $\sigma = mgu\{Z = C_z\}$

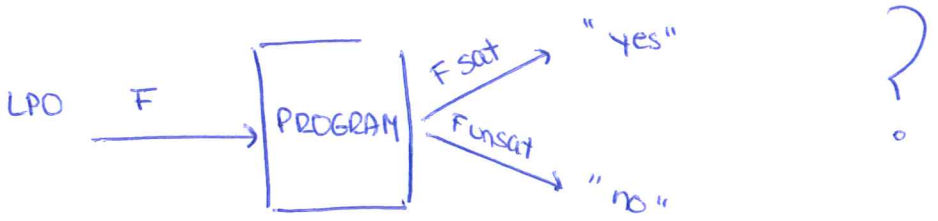
$\neg mouse(C_z)$

H) Res(G, 5c) $\sigma = mgu\{\}$

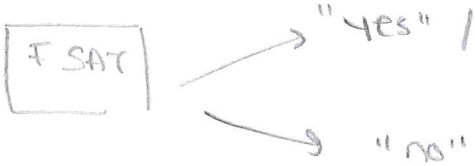
empty clause

LOGIC IN COMPUTER SCIENCE (Fall 2013)

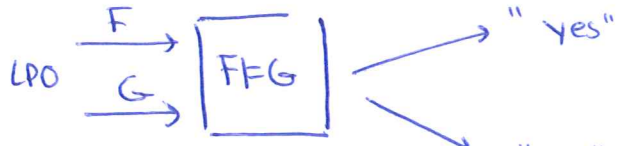
④ a) Is there any procedure such that:



No, this problem is not decidable (co-semi-decidable



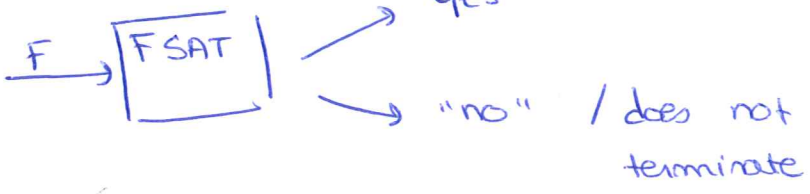
b)



Yes. Let S be the clausal form of $F \wedge \neg G$, $F \not\models G \Leftrightarrow$
$\Leftrightarrow F \wedge \neg G$ unsat $\Leftrightarrow \square \in S$.

So, the procedure computes the closure ~~under~~ of S under resolution and factoring. It will terminate as soon as it finds the empty clause but if it doesn't appear it finishes with a "no" (this may never happen).

c)



⑤ $D_I = \{0,1,2\}$ $a_I = 2$ $P_I(n,m) = 1 \Leftrightarrow m = (n+1) \bmod 3$

Let F be $\forall x \exists y \exists z\ P(x,y) \wedge (P(y,z) \vee P(z,x))$. Do we have $I \models F$?

Prove it.

$x = 2$ $\quad P_I(z,2) = 1 \Leftrightarrow 2 = (n+1) \bmod 3$

$z = 1$ $\qquad n = 1$

$x = 0$ $\quad P_I(z,0) = 1 \Leftrightarrow 0 = n+1 \bmod 3$

$n = 2$

$z = 2$

$x = 1$ $\quad P_I(z,1) = 1 \Leftrightarrow 1 = n+1 \bmod 3$

$n = 0$

$z = 0$

$P(2,y) = 1 \Leftrightarrow y = 2+1 \bmod 3$

$\boxed{y = 0}$

$P(0,z) = 1 \Leftrightarrow z = 0+1 \bmod 3$

$z = 1$

Yes, if we choose $y = 0$ and $z = 1$ for any x of D_I

b) F is $\forall x \exists y \forall z\ P(x,y,z) \wedge Q(y)$

G is $\exists y \forall x \forall z\ Q(x) \wedge P(x,y,z)$

$\Big\}$ $F \equiv G$? $F \models G$? $G \models F$?

They are not equivalent.

Because for I where $D_I = \{a,b\}$

$P_I(\ldots) = 1$ and $Q_I(a) = 1$ and $Q_I(b) = 0$

$I \models F$ but $I \not\models G$.

But $G \models F$. We can prove it by resolution.

(¬F) $\exists x \forall y \exists z\ \neg P(x,y,z) \vee \neg Q(y) \equiv \neg P(C_x, Y, C_z) \vee \neg Q(Y)$

(G) $Q(x) \wedge P(x, C_y, z)$

$P(x, C_y, z)$ $\qquad \neg P(C_x, Y, C_z) \vee \neg Q(Y)$

$\{X = C_x, Y = C_y, Z = C_z\}$ $\rightarrow$ $\dfrac{Q(C_y) \qquad Q(Y)}{\square}$ $\leftarrow X = C_y$

LOGIC IN COMPUTER SCIENCE (spring 2013)

⑥ Formalize in first order logic and prove by resolution that

$$A \wedge B \wedge C \wedge D \models E.$$

A) All owners of crazy dogs are stupid.

B) Everything has some owner.

C) There are no stupid people.

D) Pluto is a crazy dog.

E) Mourinho is very well educated.

owns(x,y)
crazydog(x)
stupid(x)

• Formalization.

A) $\forall x (\exists y \; owns(x,y) \wedge crazydog(y)) \rightarrow stupid(x))$

B) $\forall x \exists y \quad owns(y,x)$

C) $\forall x \quad \neg stupid(x)$

D) $crazydog(Pluto)$

E) $well\text{-}educated(Mourinho)$

• Clausal form.

A) $\forall x (\forall y \; \neg owns(x,y) \vee \neg crazydog(y)) \vee stupid(x)$

$\neg owns(x,y) \vee \neg crazydog(y) \vee stupid(x)$

B) $owns(f_y(z), z)$

C) $\neg stupid(v)$

D) $crazydog(Pluto)$

NOT E) $\neg welleducated(Mourinho)$

• Proof by resolution.

① $Res(A,B) \quad \sigma = mgu\{x = f_y(z), y = z\}$

$\neg crazydog(z) \vee stupid(f_y(z))$

② $Res(1,C) \quad \sigma = mgu\{v = f_y(z)\}$

$\neg crazydog(z)$

③ $Res(2,D) \quad \sigma = mgu\{z = Pluto\}$

emptyclause

LOGIC IN COMPUTER SCIENCE (Fall 2012)

⑤ A: Jack is Rob's brother.
B: Mike is Rob's brother.
C: If one person is someone's brother, then this 2nd person is also a brother of the 1st one.
D: Jack is Mike's brother.

1) Prove (by reasoning about interpretations) that $A \wedge B \wedge C \not\models D$

2) Write E such that $A \wedge B \wedge C \wedge E \models D$. and prove it by resolution.

1) A: brother(Jack, Rob)
B: brother(Mike, Rob)
C: $\forall x \forall y$ brother$(x,y) \rightarrow$ brother$(y,x) \equiv \overline{bro(x,y)} \vee bro(y,x)$
Not D: $\neg$ brother(Jack, Mike)

$bro(Jack,Rob) \wedge bro(Mike,Rob) \wedge (\forall x \forall y\ bro(x,y) \rightarrow bro(y,x)) \not\models$ bro(Jack, Mike)

we define an interpretation I such that $I \models A \wedge B \wedge C$ but $I \not\models D$. Let I be: $D_I = \{a, b\}$, $Rob_I = a$, $Jack_I = Mike_I = b$, $B_I(x,y) = (x \neq y)$

Then $I \models A \wedge B \wedge C$ but $I \not\models D$.

2) The missing property is transitivity.

E: $\forall x \forall y \forall z$ brother$(x,y) \wedge$ brother$(y,z) \rightarrow$ brother(x,z)

• Proof by resolution $\equiv \neg bro(x,y) \vee \neg bro(y,z) \vee bro(x,z)$

① Res(B, C) $\sigma = mgu\{x = Mike, y = Rob\}$
bro(Rob, Mike)

② Res(A, E) $\sigma = mgu\{x = Jack, y = Rob\}$
$\neg bro(Rob, z) \vee bro(Jack, z)$

③ Res(2, 1) $\sigma = mgu\{z = Mike\}$
bro(Jack, Mike)

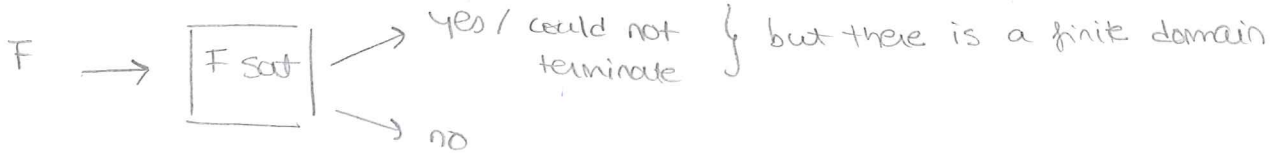
④ Res(3, D) $\sigma = \{\}$
empty clause

10

③ $D_I = \{0, 1\}$ f_I : 1-ary function symbol

P_I : 3-ary predicate symbol

a) Is it decidable whether I satisfies a given formula F? If so, what is the complexity of this?



Yes, this is decidable. If the domain is finite one can apply:

[Avaluar $\forall x$, $\exists x$ ...]

The complexity for this is high, the best well-known algorithms are exponential.

We can see that it is as hard as 3-SAT; note that a 3-SAT problem like $(\overline{x_7} \vee x_8 \vee \overline{x_2}) \wedge \ldots$ is sat $\Leftrightarrow \exists x_1 \exists x_2 \ldots \exists x_n P(f(x_3), x, f(x_2)) \wedge$ $\wedge \ldots \wedge$ evaluates true if f is interpreted as logical negation ($f_I(0) = 1$ and $f_I(1) = 0$) and $P_I(x, y, z) =$ true $\Leftrightarrow$ if at least one of its arguments is 1.

b) $D_I = \{0, 1, 2 \ldots\}$ (∞)

f: 2-ary function symbol : sum

g: " " : product

P: 1-ary predicate symbol

$P_I(0) =$ true, $P_I(x) =$ false if $x \neq 0$

Is it decidable wether I satisfies a given formula F?

No, this is undecidable. One can express well-known undecidable problems like checking whether an arbitrary integer polynomial over several variables has roots: for example $x^3y + 3z^2 + \ldots = 0$ has solutions iff $\exists x \exists y \exists z$ $P(f(f(g(g(g(x,x),x)\ldots)$ evaluates to true in this interpretation.

④ Formalize and prove by resolution $A \wedge B \wedge C \models D$.

Loves(X,Y)
mom(X)
dad(X)
Stupid(X)

A: Everybody loves his father and his mother

B: John is stupid

C: When someone is stupid, at least one of his parents is stupid too.

D: There are stupid people that are loved by someone.

A: $\forall x \; Loves(X, f(X)) \wedge Loves(X, mom(X))$

B: $Stupid(John)$

C: $\forall x \; (Stupid(X) \rightarrow (Stupid(mom(X)) \vee Stupid(dad(X))))$

D: $\exists y \exists z \; Stupid(y) \wedge Loves(z, y)$

• clausal form.

A) 1. $Loves(X, dad(X))$ 2. $Loves(X, mom(X))$

B) $Stupid(John)$

C) $\forall x \; \neg Stupid(X) \vee Stupid(mom(X)) \vee Stupid(dad(X))$

(NOT) D) $\forall y \forall z \; \neg Stupid(Y) \vee \neg Loves(Z, Y)$

• Proof by resolution.

① Res (A, D) $\sigma = \{X = Z, Y = dad(X)\}$

$\neg Stupid(dad(X))$

② Res (B, C) $\sigma = \{X = John\}$

$Stupid(mom(John)) \vee Stupid(dad(John))$

③ Res (1, 2) $\sigma = \{dad(X) = dad(John)\}$

$Stupid(mom(John))$

④ Res (3, D)

$\neg Loves(Z, mom(John))$

⑤ Res (4, 2) $\sigma = mgu\{X = Z, X = John\}$

empty clause

LOGIC IN COMPUTER SCIENCE (Fall 2011)

(4) A: All politicians sometime lie

B: Friends of football players never lie

C: Messi is a football player

D: Messi has no friends that are politicians.

A: $\forall x$ politician$(x) \rightarrow$ lies(x)

B: $\forall x \ (\exists y: \text{Friends}(x,y) \wedge \text{Player}(y)) \rightarrow \neg \text{Lies}(x)$

C: $\text{Player}(\text{Messi})$

D: $\forall x \quad \text{Friends}(x, \text{Messi}) \rightarrow \neg \text{Politician}(x)$

Not D: $\neg (\forall x \ \neg \text{Friends}(x, \text{Messi}) \vee \neg \text{Politician}(x)) \equiv$

$\exists x \quad \text{Friends}(x, \text{Messi}) \wedge \text{Politition}(x)$

- Clausal form

A) $\neg \text{Pol}(x) \vee \text{Lies}(x)$

B) $\forall x (\forall y \ \neg \text{Friends}(x,y) \vee \neg \text{Player}(y)) \vee \neg \text{Lies}(x) \equiv$

$\neg F(x,y) \vee \neg \text{Player}(y) \vee \neg \text{Lies}(x)$

C: $\text{Player}(\text{Messi})$

D: 1) $\text{Friends}(f_x(\text{Messi}), \text{Messi})$

2) $\text{Politician}(f_x(\text{Messi}))$

- Proof by resolution

(1) Res(A, D2) $\quad \sigma = \{ X = f_x(\text{Messi}) \}$

$\text{Lies}(f_x(\text{Messi}))$

(2) Res(1, B) $\quad \sigma = \{ X = f_x(\text{Messi}) \}$

$\neg \text{Friends}(f_x(\text{Messi}), Y) \vee \neg \text{Player}(Y)$

(3) Res(2, C) $\quad \sigma = \text{mgu} \{ Y = \text{Messi} \}$

$\neg \text{Friends}(f_x(\text{Messi}), \text{Messi})$

(4) Res(3, D1) $\quad \sigma = \text{mgu} \{ \}$

empty clause.

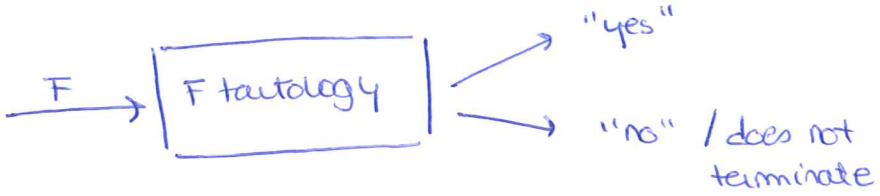
B) $\forall x \forall y \exists z \; q(x,z,y) \models \forall x \exists z \forall y \; q(x,z,y)$

(F) (G)

$q_I(x,z,y) = \text{true} \iff z == y$

$D_I = \{a, b\}$ $\quad I \models F$ but $I \not\models G$.

⑥

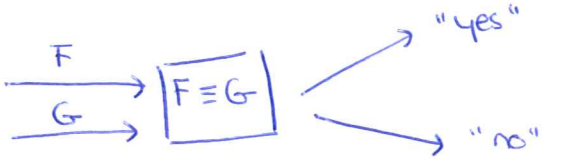


?

Yes, this is possible. This is a semi-decidable problem. To prove F tautology, we have to prove that ¬F is unsatisfiable. And, to do this we have to turn (¬F) into its closure under resolution and factoring and try to get the empty clause from it. If we find it, then the output is yes. However, we can find that the empty clause is not in S or we may never find it and ~~the~~ it doesn't terminate.

LOGIC IN COMPUTER SCIENCE (Spring 2016)

④



?

This is not possible.

This is a semi-decidable problem. To prove that $F \equiv G$:

① Convert $(F \wedge \neg G) \vee (\neg F \wedge G)$ into its clausal form S_0.
We have $F \equiv G$ if S_0 is unsat.

② Compute the closure under resolution and factoring of S_0

2.1.- If we find the empty clause in S_i $\Rightarrow$ return "yes"

2.2.- Otherwise, obtain S_{i+1} by resolution and factoring of S_i

- If there is no new clause $\Longrightarrow$ return "no"
- Else $\Rightarrow$ goto 2.1 with the next i.

LOGIC IN COMPUTER SCIENCE (Spring 2017)

④ ① bro(joan, pere)

② father(enric, joan)

③ $\forall N \forall U$ (uncle(N,U) $\leftarrow \exists F$ (father(N,F) $\wedge$ brother(F,U)))

- clausal form.

③ $\forall N \forall U$ ($\forall F$ $\neg$father(N,F) $\vee \neg$brother(F,U)) $\vee$ uncle(N,U))

$\equiv \neg fa(N,F) \vee \neg bro(F,U) \vee unc(N,U)$

- $\exists x \exists y$ uncle(x,y) log consequence of P.

↳ ④ $\neg(\exists x \exists y$ uncle(x,y)) $\equiv \forall x \forall y$ $\neg$uncle(x,y)

clausal form → $\neg$uncle(x,y)

- Proof by resolution

(A) Res(4,3) σ = mgu {N=X, U=Y}

$\neg fa(X,F) \vee \neg bro(F,Y)$

(B) Res(A,2) σ = mgu {X=enric, F=joan}

$\neg bro(joan, Y)$

(C) Res(B,1) σ = mgu {Y=pere}

empty clause

⑤ True or false?

A). $\forall x \exists y$ $(p(x, f(y)) \wedge \neg p(x,y))$ is sat

Let I be an interpretation such that

$D_I = \{a, b\}$

$f_I(a) = b$

$f_I(b) = a$

p(x,y)
1
0
0
1

x	y
a	a
a	b
b	a
b	b

Therefore, it's true that this is satisfiable.

- Demuestra que la skolemización no preserva la equivalencia lógica:

Definimos una interpretación I con dominio $D_I = \{a, b\}$, una función $p_I: D_I \times D_I \to \{0,1\}$ tal que $p_I(a,a) = 1$ $p_I(a,b) = 0$ $b_I(b,b) = 1$ $p_I(b,a) = 0$

una función:
$f_I : D_I \to D_I$ tal que $f_I(a) = b$ y $f_I(b) = a$

- Podemos comprobar que $I \not\models \forall x \exists y\ p(x,y)$ pero que $I \not\models \forall x\ p(x, f(x))$ ⟹ Ambas fórmulas no son lógicamente equivalentes. ← la transformación con skolemización.

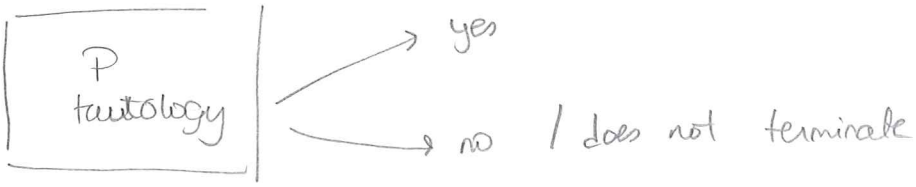
- Demuestra que la resolución sin la factorización no es refutacionalmente completa.
  - El conjunto de cláusulas $S = \{ p(x) \vee p(y),\ \neg p(w) \vee \neg p(z) \}$ es insat.
  - Si aplicamos solo resolución, lo único que podemos obtener de nuevo es $p(x_3) \vee \neg p(z_3)$, entonces no conseguimos la cláusula vacía.

S insat $\Leftrightarrow \square \in Res(S)$

- Demuestra que la resolución unitaria termina para S (conjunto de cláusulas de Horn). ≡ la clausura bajo resolución es finita.

K = núm de literales de la cláusula de mayor número de literales de S, K también lo es para $ResUnit(S)$.

A = aridad del símbolo de predicado de aridad máxima que aparece en S. Entonces en $ResUnit(S)$ no puede aparecer una cláusula de más de $n = A \cdot K$ variables. Las variables de cada cláusula de $ResUnit(S)$ pueden escribirse como el conjunto $\{x_1 \dots x_m\}$ y, por lo tanto, los átomos de las cláusulas de $ResUnit(S)$ serán un símbolo de predicado aplicado a variables de $\{x_1 \dots x_n\}$ y constantes de las que aparecen en S. Como solo hay un número finito de átomos de este tipo, el número de cláusulas distintas que pueden aparecer en $ResUnit(S)$ también es finito, por lo que la clausura bajo res. es finita.

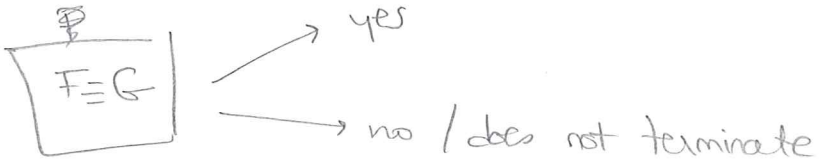


This is possible, in fact this is semi-decidable

P's algorithm.

(1) we find the empty clause $\Longrightarrow$ yes

(2) There isn't an empty clause and ResFact(S) is finite $\Longrightarrow$ no
Does not terminate.



(1) $F \equiv G \iff$ clausal form S_0 of $(F \wedge \neg G) \vee (\neg F \wedge G)$ unsat

(2) Compute the closure under resolution and factoring of S_0 by levels:

A) $\square \in S_i \Rightarrow F \equiv G \Rightarrow$ yes

B) Obtain S_{i+1} by adding all clauses obtained by one step of ResFact on clauses in S_i

C) No new clause $\Rightarrow$ no
otherwise $\Rightarrow$ goto A.

Undecidable. We know that given a quadratic equation with just one variable, it is decidable to find if it has any integer solution. However, this is undecidable for any polynomials equations $P = 0$ if P is any polynomial with products between variables.
It would be decidable if we could evaluate arbitrary formulas in the integers.

We can easily express any polynomial such that $T_P =$ sum (prod (sum ...)

such that $P = 0$ has no integer solution $\iff \forall x \forall y \forall z \ldots \neg eq_I(T_P, zero)$

where D_I are the integers and eq_I, sum_I, $prod_I$, $zero_I$ are

the functions $=, 0, *, +1$.

- Demuestra que la satisfactibilidad de conjuntos de Horn S sin símbolos de función de aridad mayor que cero es decidible.

Sabemos: - la resolución unitaria es correcta y refutacionalmente completa para cláusulas de Horn

⟹ Entonces: la satisfactibilidad de conjuntos de Horn S sin símbolos de función de aridad mayor que cero es decidible ⟹ se obtiene ResUnit(S) en tiempo finito, y tenemos $\square \in ResUnit(S) \iff S$ insat.

$F \Rightarrow$ símbolos de función

$P \Rightarrow$ símbolos de predicado

$X \Rightarrow$ símbolos de variable

término
- variables
- const = (sí. función)°
- $f(t_1 \dots t_n)$ $t_1 \dots t_n$ términos $n > 0$

átomo
- $p(t_1 \dots t_n)$ $t_1 \dots t_n$ términos $n > 0$
- (sí. predicado)°

Todo átomo es una fórmula.

- Regla
  - correcta: sólo cons. lógicas
  - completa: todas cons. lógicas

$P = \{p^2, q^1\}$ $F = \{f^2, g^1, a^0, b^0, c^0\}$

≠ átomos?

átomo $\begin{cases} p(t_1 \dots t_n) \\ \underbrace{\quad}_{\text{términos}} \quad n>0 \\ (\text{símbolo de pred})^0 \end{cases}$

potencialmente infinitos

(A) $\forall x \; esfeliz(x) \leftarrow (\forall Y \; alumnode(X,Y) \rightarrow analógica(Y))$

$\equiv (\exists Y \; \neg(alumno(X,Y) \rightarrow analógica(Y)) \vee \forall x \; esfeliz(x)$

$\equiv \exists Y \; \neg(\neg alumno(X,Y) \vee analógica(Y)) \vee \forall x \; esfeliz(x)$

$\equiv (\exists Y \; alumno(X,Y) \wedge \neg analógica(Y)) \vee \forall x \; esfeliz(x)$

$\equiv (alumno(X, f_Y(X)) \wedge \neg analógica(f_Y(X)) \vee esfeliz(x)$

(A1) (A2)

(B) $\forall x \; (esfeliz(x) \leftarrow \neg \exists Y \; estudiante(X,Y))$

(¬B) $\neg(\forall x \; (esfeliz(x) \vee \neg(\neg \exists Y \; estudiante(X,Y))))$

$\exists X \; \neg esfeliz(X) \wedge \neg \exists Y \; estudiante(X,Y)$

$\exists X \; \neg esfeliz(X) \wedge \forall Y \; \neg estudiante(X,Y)$

$\neg esfeliz(C_X) \wedge \neg estudiante(C_X, Y)$

(B1) (B2)

---

① • $Res(A1, B1)$ $\sigma = \{X = C_X\}$

$alumno(C_X, f_Y(C_X))$

② • $Res(1, B2)$ $\sigma = \{Y = f_Y(C_X)\}$

□

$\forall P \forall Y \; (alumno(P,Y) \rightarrow analog(Y)) \rightarrow esfeliz(P)$