On the Complexity of Negative Quantification

Aleksy Schubert, Paweł Urzyczyn, Konrad Zdanowski

Types 2014, Paris, May 14th, 2014

Prenex normal form

In classical logic:

Every formula is classically equivalent to one of the form:

$$Q_1 x_1 Q_2 x_2 \ldots Q_k x_k . \mathit{Body}(x, x_2, \ldots, x_k),$$

where *Body* has no quantifiers.

Classification:

Formulas may be classified according to the quantifier prefix, e.g. universal ($\forall^*$) formulas are in Π_1, and Π_2 is $\forall^*\exists^*$, etc.

In intuitionistic logic:

The prenex fragment is decidable in Pspace.

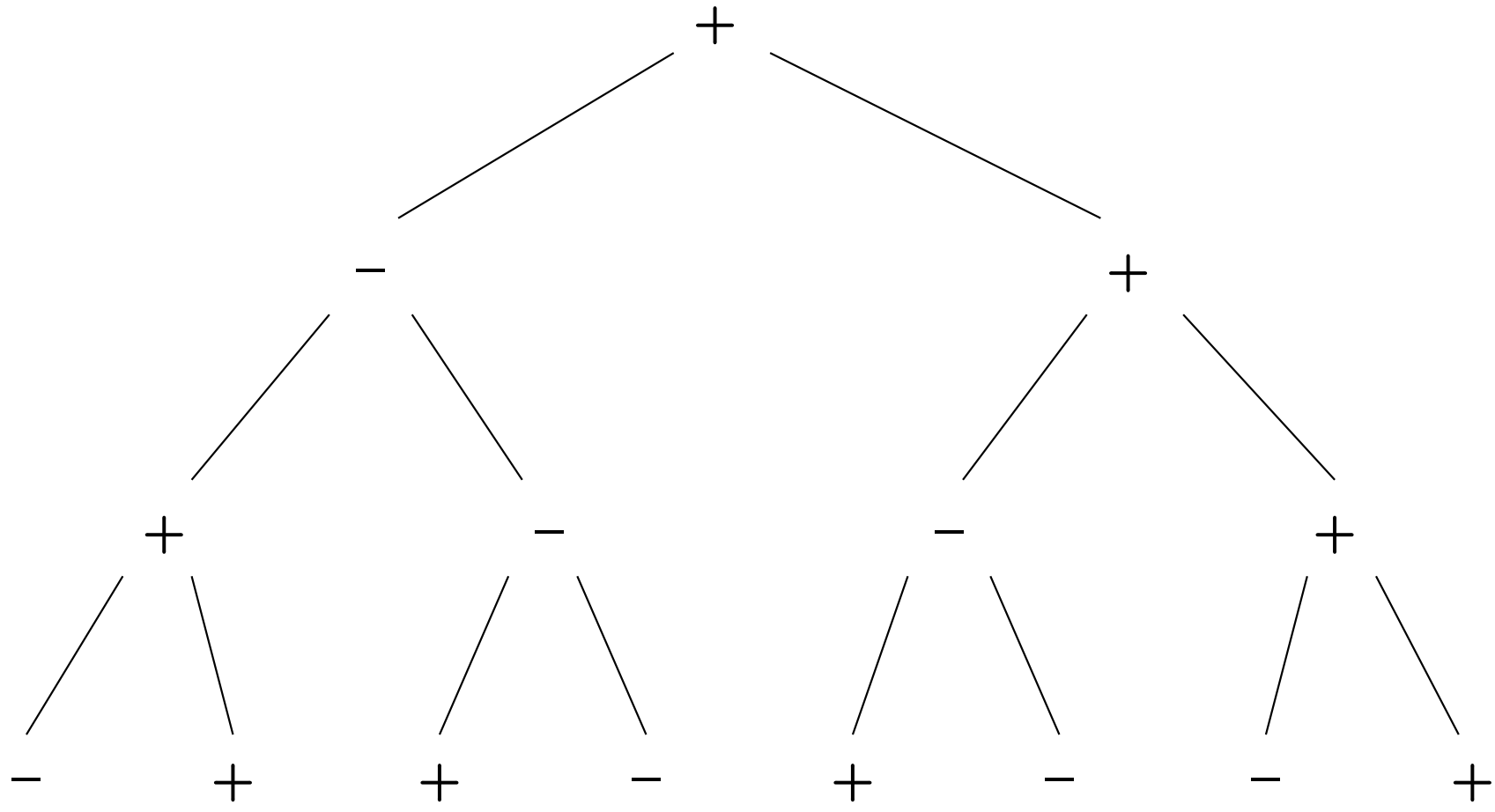
Mints Hierarchy

Can we restore the prenex classification in intuitionistic logic?

Grigori Mints (1968): Yes, consider the quantifier prefix
a formula *would get*, if classically normalized.
(What counts is the alternation of quantifiers.)

For example, $\forall$ quantifiers occurring at *positive* positions
will remain $\forall$ in the prefix.

Positive and Negative



Mints Hierarchy

Π_1 – All quantifiers at positive positions.

Σ_1 – All quantifiers at negative positions.

Π_2 – One alternation: some negative quantifiers in scope of some positive ones.

Σ_2 – One alternation: some positive quantifiers in scope of some negative ones.

And so on.

The language we study

To make things simpler, we consider first-order formulas

- with universal quantifiers and implications;
- without function symbols.

This fragment is known to be undecidable.

Examples

$(\forall x\, P(x)) \to Q$	is a Σ_1 (negative) formula;
$((\forall x\, P(x)) \to Q) \to R$	is a Π_1 (positive) formula;
$(\forall x\, (\forall y\, P(x) \to \forall z\, S(y, z))) \to R$	is a Σ_2 formula.

$\circ$

Complexity of Mints Hierarchy

- Π_2 and Σ_2 are undecidable.
- Π_1 is **co-2-NExptime**-hard
  (conjecture: super-elementary)
- Σ_1 (this talk):
  - **Expspace**-complete in general;
  - **co-Nexptime**-complete with monadic predicates.

Σ_1 decision problem

Negative formulas are of shape

$$\varphi_1 \to \cdots \to \varphi_n \to \sigma,$$

where σ is an atomic formula and $\varphi_1, \ldots, \varphi_n$ are positive.

A Σ_1 decision problem can be seen as

$$\varphi_1, \ldots, \varphi_n \vdash \sigma$$

(Derive an atom from positive assumptions.)

Upper bound for Σ_1

Proofs in Σ_1 may use universal assumptions, which can be instantiated. But introducing new variables is of no use.

If there is a proof then there is one where all eigenvariables are free variables of the original formula φ.

Such proof can be constructed in exponential space.
(Every subformula of φ of size n has at most n^n instances.)

Towards the lower bound

Assumptions:

$\forall xyz\,(P(x,y,z,1) \to P(x,y,z,0))$,

$\forall xy\,(P(x,y,1,0) \to P(x,y,0,1))$,

$\forall x\,(P(x,1,0,0) \to P(x,0,1,1))$,

$P(1,0,0,0) \to P(0,1,1,1)$.

Goal: $P(0,0,0,0)$

Towards the lower bound

Assumptions:

$\forall xyz\,(P(x,y,z,1) \to P(x,y,z,0))$,

$\forall xy\,(P(x,y,1,0) \to P(x,y,0,1))$,

$\forall x\,(P(x,1,0,0) \to P(x,0,1,1))$,

$P(1,0,0,0) \to P(0,1,1,1)$.

Goal: $P(0,0,0,1)$

Towards the lower bound

Assumptions:

$\forall xyz\,(P(x,y,z,1) \to P(x,y,z,0))$,

$\forall xy\,(P(x,y,1,0) \to P(x,y,0,1))$,

$\forall x\,(P(x,1,0,0) \to P(x,0,1,1))$,

$P(1,0,0,0) \to P(0,1,1,1)$.

Goal: $P(0,0,1,0)$

Towards the lower bound

Assumptions:

$\forall xyz\,(P(x,y,z,1) \to P(x,y,z,0))$,

$\forall xy\,(P(x,y,1,0) \to P(x,y,0,1))$,

$\forall x\,(P(x,1,0,0) \to P(x,0,1,1))$,

$P(1,0,0,0) \to P(0,1,1,1)$.

Goal: $P(0,0,1,1)$

Towards the lower bound

Assumptions:

$\forall xyz\,(P(x,y,z,1) \to P(x,y,z,0))$,

$\forall xy\,(P(x,y,1,0) \to P(x,y,0,1))$,

$\forall x\,(P(x,1,0,0) \to P(x,0,1,1))$,

$P(1,0,0,0) \to P(0,1,1,1)$.

Goal: $P(0,1,0,0)$

Towards the lower bound

Assumptions:

$\forall xyz\,(P(x,y,z,1) \to P(x,y,z,0))$,

$\forall xy\,(P(x,y,1,0) \to P(x,y,0,1))$,

$\forall x\,(P(x,1,0,0) \to P(x,0,1,1))$,

$P(1,0,0,0) \to P(0,1,1,1)$.

Goal: $P(0,1,0,1)$

Towards the lower bound

Assumptions:

$\forall xyz\,(P(x,y,z,1) \to P(x,y,z,0))$,

$\forall xy\,(P(x,y,1,0) \to P(x,y,0,1))$,

$\forall x\,(P(x,1,0,0) \to P(x,0,1,1))$,

$P(1,0,0,0) \to P(0,1,1,1)$.

Goal: a few more steps. . .

Towards the lower bound

Assumptions:

$\forall xyz\,(P(x,y,z,1) \to P(x,y,z,0))$,

$\forall xy\,(P(x,y,1,0) \to P(x,y,0,1))$,

$\forall x\,(P(x,1,0,0) \to P(x,0,1,1))$,

$P(1,0,0,0) \to P(0,1,1,1)$.

Goal: $P(1,1,1,1)$

Towards the lower bound

Assumptions:

$\forall xyz\,(P(x,y,z,1) \to P(x,y,z,0))$,

$\forall xy\,(P(x,y,1,0) \to P(x,y,0,1))$,

$\forall x\,(P(x,1,0,0) \to P(x,0,1,1))$,

$P(1,0,0,0) \to P(0,1,1,1)$.

Goal:

The proof search is as difficult as rewriting $\mathbf{0000}$ into $\mathbf{1111}$.

Another example:

$\forall xyzx'y'z'\,(D(x,x') \to D(y,y') \to D(z,z') \to$
$((P(x',y',z',3) \to P(x',y',z',2)) \to P(x,y,z,1)) \to P(x,y,z,0))$

$\forall xyx'y'\,(D(x,x') \to D(y,y') \to$
$((P(x',y',3,2) \to P(x',y',2,3)) \to P(x,y,1,0)) \to P(x,y,0,1))$

$\forall xx'\,(D(x,x') \to$
$((P(x',3,2,2) \to P(x',2,3,3)) \to P(x,1,0,0)) \to P(x,0,1,1))$

$(P(3,2,2,2) \to (2,3,3,3)) \to P(1,0,0,0) \to P(0,1,1,1),$

$P(2,2,2,2) \to P(1,1,1,1),\ P(3,3,3,3),\ D(2,0),\ D(3,1).$

Goal: $P(0,0,0,0)$

Another example:

$\forall xyzx'y'z'\,(D(x,x') \to D(y,y') \to D(z,z') \to$
$((P(x',y',z',3) \to P(x',y',z',2)) \to P(x,y,z,1)) \to P(x,y,z,0))$

$\forall xyx'y'\,(D(x,x') \to D(y,y') \to$
$((P(x',y',3,2) \to P(x',y',2,3)) \to P(x,y,1,0)) \to P(x,y,0,1))$

$\forall xx'\,(D(x,x') \to$
$((P(x',3,2,2) \to P(x',2,3,3)) \to P(x,1,0,0)) \to P(x,0,1,1))$

$(P(3,2,2,2) \to (2,3,3,3)) \to P(1,0,0,0) \to P(0,1,1,1),$

$P(2,2,2,2) \to P(1,1,1,1),\ P(3,3,3,3),\ D(2,0),\ D(3,1).$

Goal: $P(0,0,0,1)$

New assumptions: $P(2,2,2,3) \to P(2,2,2,2),$

Another example:

$\forall xyzx'y'z'\,(D(x,x') \to D(y,y') \to D(z,z') \to$
$((P(x',y',z',3) \to P(x',y',z',2)) \to P(x,y,z,1)) \to P(x,y,z,0))$

$\forall xyx'y'\,(D(x,x') \to D(y,y') \to$
$((P(x',y',3,2) \to P(x',y',2,3)) \to P(x,y,1,0)) \to P(x,y,0,1))$

$\forall xx'\,(D(x,x') \to$
$((P(x',3,2,2) \to P(x',2,3,3)) \to P(x,1,0,0)) \to P(x,0,1,1))$

$(P(3,2,2,2) \to (2,3,3,3)) \to P(1,0,0,0) \to P(0,1,1,1),$

$P(2,2,2,2) \to P(1,1,1,1),\ P(3,3,3,3),\ D(2,0),\ D(3,1).$

Goal: $P(0,0,1,0)$

New assumptions: $P(2,2,2,3) \to P(2,2,2,2),$
$P(2,2,3,2) \to P(2,2,2,3),$

Another example:

$\forall xyzx'y'z'\,(D(x,x') \to D(y,y') \to D(z,z') \to$
$((P(x',y',z',3) \to P(x',y',z',2)) \to P(x,y,z,1)) \to P(x,y,z,0))$

$\forall xyx'y'\,(D(x,x') \to D(y,y') \to$
$((P(x',y',3,2) \to P(x',y',2,3)) \to P(x,y,1,0)) \to P(x,y,0,1))$

$\forall xx'\,(D(x,x') \to$
$((P(x',3,2,2) \to P(x',2,3,3)) \to P(x,1,0,0)) \to P(x,0,1,1))$

$(P(3,2,2,2) \to (2,3,3,3)) \to P(1,0,0,0) \to P(0,1,1,1),$

$P(2,2,2,2) \to P(1,1,1,1),\ P(3,3,3,3),\ D(2,0),\ D(3,1).$

Goal: $P(0,0,1,1)$

New assumptions: $P(2,2,2,3) \to P(2,2,2,2),$
$P(2,2,3,2) \to P(2,2,2,3),\ P(2,3,2,2) \to P(3,2,2,2),$

Another example:

$\forall xyzx'y'z'\,(D(x,x') \to D(y,y') \to D(z,z') \to$
$((P(x',y',z',3) \to P(x',y',z',2)) \to P(x,y,z,1)) \to P(x,y,z,0))$

$\forall xyx'y'\,(D(x,x') \to D(y,y') \to$
$((P(x',y',3,2) \to P(x',y',2,3)) \to P(x,y,1,0)) \to P(x,y,0,1))$

$\forall xx'\,(D(x,x') \to$
$((P(x',3,2,2) \to P(x',2,3,3)) \to P(x,1,0,0)) \to P(x,0,1,1))$

$(P(3,2,2,2) \to (2,3,3,3)) \to P(1,0,0,0) \to P(0,1,1,1),$

$P(2,2,2,2) \to P(1,1,1,1),\ P(3,3,3,3),\ D(2,0),\ D(3,1).$

Goal: $P(0,1,0,0)$

New assumptions: $P(2,2,2,3) \to P(2,2,2,2),$
$P(2,2,3,2) \to P(2,2,2,3),\ P(2,3,2,2) \to P(3,2,2,2),$
$P(3,2,2,2) \to P(2,3,3,3),$

Another example:

$\forall xyzx'y'z'\,(D(x,x') \to D(y,y') \to D(z,z') \to$
$((P(x',y',z',3) \to P(x',y',z',2)) \to P(x,y,z,1)) \to P(x,y,z,0))$

$\forall xyx'y'\,(D(x,x') \to D(y,y') \to$
$((P(x',y',3,2) \to P(x',y',2,3)) \to P(x,y,1,0)) \to P(x,y,0,1))$

$\forall xx'\,(D(x,x') \to$
$((P(x',3,2,2) \to P(x',2,3,3)) \to P(x,1,0,0)) \to P(x,0,1,1))$

$(P(3,2,2,2) \to (2,3,3,3)) \to P(1,0,0,0) \to P(0,1,1,1),$

$P(2,2,2,2) \to P(1,1,1,1),\ P(3,3,3,3),\ D(2,0),\ D(3,1).$

Goal: $P(0,1,0,1)$

New assumptions: $P(2,2,2,3) \to P(2,2,2,2),$
$P(2,2,3,2) \to P(2,2,2,3),\ P(2,3,2,2) \to P(3,2,2,2),$
$P(3,2,2,2) \to P(2,3,3,3),\ P(3,2,2,3) \to P(3,2,2,2),$

Another example:

$\forall xyzx'y'z'\,(D(x,x') \to D(y,y') \to D(z,z') \to$
$((P(x',y',z',3) \to P(x',y',z',2)) \to P(x,y,z,1)) \to P(x,y,z,0))$

$\forall xyx'y'\,(D(x,x') \to D(y,y') \to$
$((P(x',y',3,2) \to P(x',y',2,3)) \to P(x,y,1,0)) \to P(x,y,0,1))$

$\forall xx'\,(D(x,x') \to$
$((P(x',3,2,2) \to P(x',2,3,3)) \to P(x,1,0,0)) \to P(x,0,1,1))$

$(P(3,2,2,2) \to (2,3,3,3)) \to P(1,0,0,0) \to P(0,1,1,1),$

$P(2,2,2,2) \to P(1,1,1,1),\ P(3,3,3,3),\ D(2,0),\ D(3,1).$

Goal: a few more steps. . .

New assumptions: $P(2,2,2,3) \to P(2,2,2,2)$,
$P(2,2,3,2) \to P(2,2,2,3),\ P(2,3,2,2) \to P(3,2,2,2)$,
$P(3,2,2,2) \to P(2,3,3,3),\ P(3,2,2,3) \to P(3,2,2,2)$,

. . . . . ,

Another example:

$\forall xyzx'y'z'\,(D(x,x') \to D(y,y') \to D(z,z') \to$
$((P(x',y',z',3) \to P(x',y',z',2)) \to P(x,y,z,1)) \to P(x,y,z,0))$

$\forall xyx'y'\,(D(x,x') \to D(y,y') \to$
$((P(x',y',3,2) \to P(x',y',2,3)) \to P(x,y,1,0)) \to P(x,y,0,1))$

$\forall xx'\,(D(x,x') \to$
$((P(x',3,2,2) \to P(x',2,3,3)) \to P(x,1,0,0)) \to P(x,0,1,1))$

$(P(3,2,2,2) \to (2,3,3,3)) \to P(1,0,0,0) \to P(0,1,1,1),$

$P(2,2,2,2) \to P(1,1,1,1),\ P(3,3,3,3),\ D(2,0),\ D(3,1).$

Goal: $P(1,1,1,1)$

New assumptions: $P(2,2,2,3) \to P(2,2,2,2),$
$P(2,2,3,2) \to P(2,2,2,3),\ P(2,3,2,2) \to P(3,2,2,2),$
$P(3,2,2,2) \to P(2,3,3,3),\ P(3,2,2,3) \to P(3,2,2,2),$
$\ldots\ldots\ldots,\ P(3,3,3,3) \to P(3,3,3,2)$

Σ_1 and **Expspace**

Theorem

The problem to decide if a given Σ_1 formula is derivable in intuitionistic predicate calculus is **Expspace***-complete.*

Unary and multi-ary predicates (a diggression)

Can we translate this example to a monadic signature?

Not really: it is essential that the goals to prove represent strings being rewritten.

What counts are the targets of non-atomic formulas.
More precisely: if $\Gamma \vdash \sigma$ is a decision problem where

– all targets of non-atomic formulas in Γ are unary or nullary,

then one can find an equivalent monadic problem $\Gamma' \vdash \sigma'$.

Alternation

Existential choice
because there may be more than one usable assumption.

Universal choice
because an assumption may have more than one premise.

(To derive P from $\varphi \to \psi \to P$
one has to prove both φ and ψ.)

The monadic case

Theorem: *The decision problem for* Σ_1 *with at most unary predicates is* co-Nexptime-*complete.*

Proof:

1. A Σ_1 formula can express the halting problem for a *Branching Turing Machine* (alternating Turing Machine without existential states) working in exponential time.
2. A non-provable Σ_1 formula has an exponential size refutation.

Branching Turing Machine (BTM)

A branching Turing Machine only has deterministic and universal states. Its computation is a tree which may consist of many branches, but there is only one computation for any given input.

This means that a BTM is like a deterministic tree automaton.

Yet differently: the problem of finding an accepting computation of a BTM has at most one solution.

Branching Turing Machine (BTM)

A computation of BTM is a tree of machine configurations, each configuration consisting of a number of tape cells.

It can be seen as a finite graph of tape cells.
For a given input this graph is unique.

Such a graph can be reconstructed using a nondeterministic loop (after constructing an initial node):

1. Add a new node n to the graph;
2. Nondeterministically set the label of n;
3. Verify the correctness of the label;
4. If n is the last node then stop else go to 1.

Simulating a BTM

Reconstruction of a computation of BTM:

1. Add a new node n to the graph;
2. Nondeterministically set the label of n;
3. Verify the correctness of the guess;
4. If n is the last node then stop else go to 1.

This can be simulated by a proof-construction process for a Σ_1 decision problem, where all non-atomic formulas have the same nullary target *loop*.

With no multi-ary targets, the decision problem can be designed as a unary decision problem.

Monadic lower bound

The decision problem for monadic Σ_1 is hard for the class of languages recognized by exponential time bounded branching Turing machines. In other words, it is co-Nexptime-hard.

NB: the positive assumption formulas use in the coding are actually universal formulas (all quantifiers occur at the beginning).

Towards monadic upper bound

The process of deciding a judgment $\Gamma \vdash \phi$ can be seen as a game between an existential player E, trying to prove the judgment, and a universal player A, trying to refute it.

Player E attempts to construct a long normal proof. At every step he chooses an assumption $\psi \in \Gamma$ to be used in the next step. Player A then determines the next goal, choosing one of the premises of ψ. Player E wins when A has no choice.

Example

For example, suppose in position $\Gamma \vdash P(x)$ player E chooses the formula $\forall y((R(z) \to Q(y)) \to R(y) \to P(y))$ from Γ.

Then player A chooses the next position.
It is either $\Gamma \vdash R(x)$, or $\Gamma, R(z) \vdash Q(x)$.

$\circ$

Refutations

The game is determined. One of the players must have a strategy: there is either a proof or a refutation.

What is a refutation?

It is an infinite tree of judgments representing a strategy of player A. Every node has as many children as there are possible choices of E. Each child represents a move of A in response to some choice of E.

The size of a refutation

- If a judgment $\Gamma \vdash \sigma$ occurs in a refutation of an initial formula of size n then the size of Γ is exponential in n.
- There is at most n^2 possible choices of σ.
- Every node has at most exponentially many children.
- A refutation is infinite but can be trimmed to a finite tree, of at most exponential depth...
- ...with only polynomial number of branchings.

Conclusions

- The decision problem for Σ_1 is Exspspace-complete.
- Important: only free variables in the original formula are used.
- Important: the number of targets is exponential.
- The decision problem for Σ_1 with at most unary predicates is co-Nexptime-complete.
- Important: small number of targets.