Why propositional quantification makes modal logics on trees robustly hard ?

(joint paper with Stéphane Demri from CNRS)

Bartosz Bednarczyk

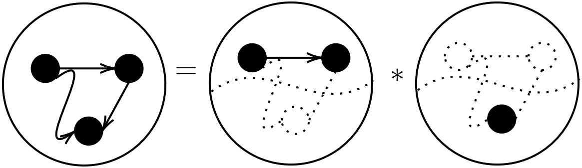
bartosz.bednarczyk@cs.uni.wroc.pl

Technische Universität Dresden
and University of Wrocław

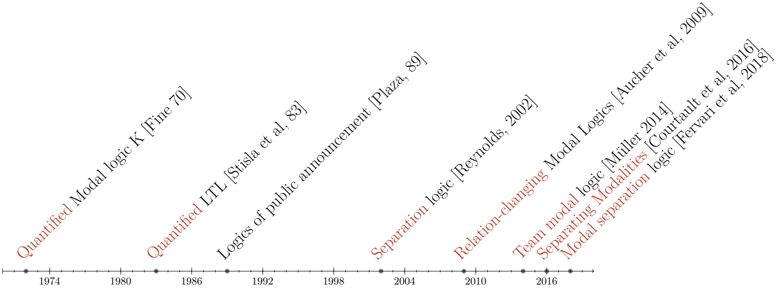
LICS 2019

Vancouver, June 26th, 2019

A concept of separation

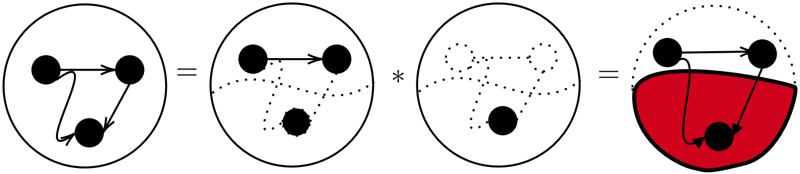


A few examples:



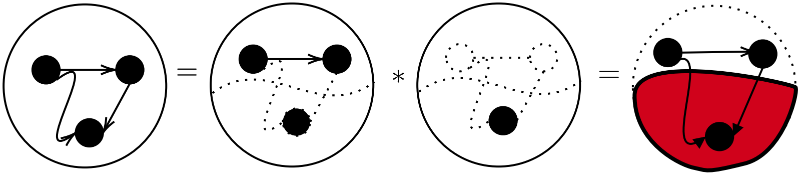
Propositional quantification - a more general setting

- Separation ≈ colouring parts with different colours



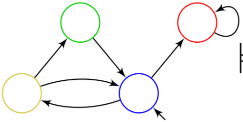
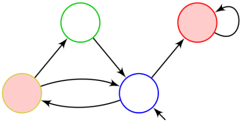
Propositional quantification - a more general setting

- Separation $\approx$ colouring parts with different colours



- Propositional quantification

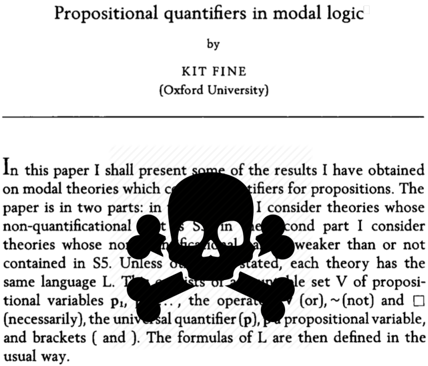
$$\mathfrak{M} \models \exists \bigcirc . \varphi \stackrel{\text{def}}{=} \varphi \text{ is satisfied after colouring } \mathfrak{M} \text{ with } \bigcirc$$

 $\models \exists \bigcirc . \varphi$ $\Leftrightarrow$  $\models \varphi$

Propositional Quantification = undecidability since 1970

- Modal logics: K, S4, GL.
- Temporal logics: QLTL, QCTL
- and even more...

The end of the story?



Propositional quantifiers in modal logic

by

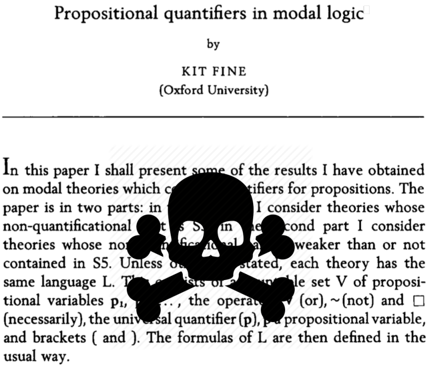
KIT FINE
(Oxford University)

In this paper I shall present some of the results I have obtained on modal theories which c[illegible]tifiers for propositions. The paper is in two parts: in [illegible] I consider theories whose non-quantificational [illegible] in [illegible]cond part I consider theories whose no[illegible] weaker than or not contained in S5. Unless o[illegible]stated, each theory has the same language L. T[illegible]le set V of propositional variables p_1, [illegible], the operat[illegible] (or), ~ (not) and □ (necessarily), the universal quantifier (p), p a propositional variable, and brackets (and). The formulas of L are then defined in the usual way.

Propositional Quantification = undecidability since 1970

- Modal logics: K, S4, GL.
- Temporal logics: QLTL, QCTL
- and even more...

The end of the story?


Propositional quantifiers in modal logic

by

KIT FINE
(Oxford University)

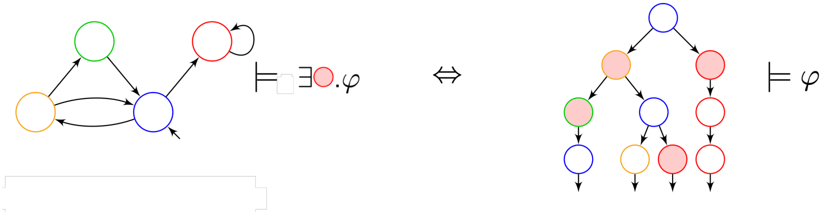
In this paper I shall present some of the results I have obtained on modal theories which c[illegible]tifiers for propositions. The paper is in two parts: in [illegible] I consider theories whose non-quantificational [illegible] S5. In [illegible]cond part I consider theories whose no[illegible] weaker than or not contained in S5. Unless o[illegible]stated, each theory has the same language L. Th[illegible]ists of a [illegible]le set V of propositional variables p_1, [illegible] . . . , the operat[illegible] ∨ (or), ~ (not) and □ (necessarily), the universal quantifier (p), p a propositional variable, and brackets (and). The formulas of L are then defined in the usual way.

Not really. Consider trees as models!

Tree semantics: the cure for undecidability

The cure for undecidability

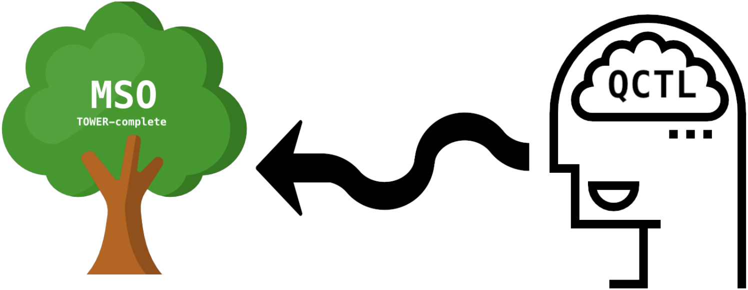
Instead of colouring models colour its tree unfolding!



Tree semantics: the cure for undecidability

Decidability on trees [Sistla et al, 87], [Laroussinie et al, 14]

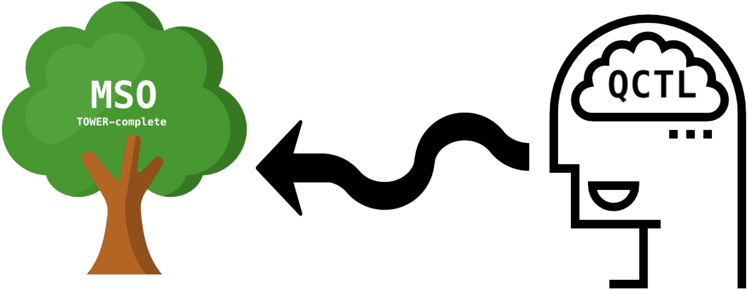
QCTL, QCTL* and QLTL on trees are TOWER-complete.



Tree semantics: the cure for undecidability

Decidability on trees [Sistla et al, 87], [Laroussinie et al, 14]

QCTL, QCTL* and QLTL on trees are TOWER-complete.



But what about standard modal logics? THIS TALK!

The main goal of this paper

The main question in this talk

What is the exact complexity of quantified MLs on trees?

TOWER-hardness for previous logics required until operator.

The main goal of this paper

The main question in this talk

What is the exact complexity of quantified MLs on trees?

TOWER-hardness for previous logics required until operator.

So maybe modal logics are elementarily decidable?

The main goal of this paper

The main question in this talk

What is the exact complexity of quantified MLs on trees?

TOWER-hardness for previous logics required until operator.

So maybe modal logics are elementarily decidable?

The answer (unpleasant truth)

Quantified standard MLs on trees are TOWER-complete.

The main goal of this paper

The main question in this talk

What is the exact complexity of quantified MLs on trees?

TOWER-hardness for previous logics required until operator.

So maybe modal logics are elementarly decidable?

The answer (unpleasant truth)

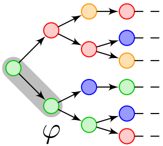
Quantified standard MLs on trees are TOWER-complete.

We sketch the hardness proof for $\mathrm{QCTL}_{\mathrm{EX}} \approx \mathrm{QK}$.

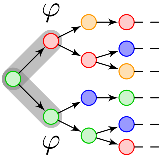
Quantified Computation-Tree Logic with **X** only

- **atomic propositions:** , , …
- **boolean combinators:** $\neg\varphi$, $\varphi \vee \psi$, $\varphi \wedge \psi$, …
- **modalities:**

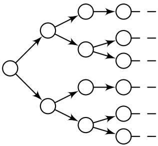
E X φ



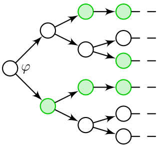
A X φ



- **propositional quantifiers:**



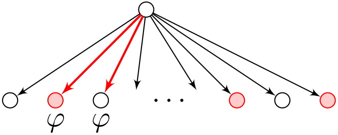
$\exists$.φ



Expressivity example: uniqueness

Non-uniqueness

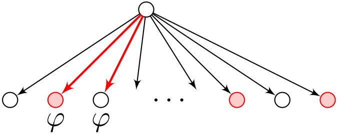
$\exists \bigcirc.\,(\mathrm{EX}(\bigcirc \land \varphi) \land \mathrm{EX}(\neg \bigcirc \land \varphi))$



Expressivity example: uniqueness

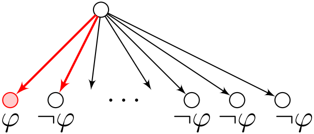
Non-uniqueness

$\exists \bigcirc. \, (\mathrm{EX}(\bigcirc \wedge \varphi) \wedge \mathrm{EX}(\neg \bigcirc \wedge \varphi))$



Uniqueness

$\mathrm{EX}(\varphi) \wedge \neg \exists \bigcirc. \, (\mathrm{EX}(\bigcirc \wedge \varphi) \wedge \mathrm{EX}(\neg \bigcirc \wedge \varphi))$



A notion of local nominals

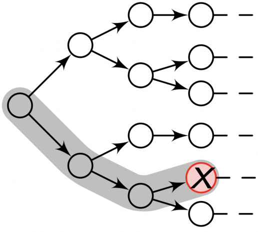
- Uniqueness expressible but only in the limited scope
- Local nominal = nominal but in limited scope

A notion of local nominals

- Uniqueness expressible but only in the limited scope
- Local nominal = nominal but in limited scope
- Useful operators: $\mathrm{nom}(\mathrm{x}, \mathrm{lvl})$ (binder) and $@_{\mathrm{x}}^{\mathrm{lvl}}\varphi$ (at).

$$\mathrm{nom}(\mathrm{x}, \mathrm{lvl}) = \mathrm{EX}_{=1}^{\mathrm{lvl}}(\mathrm{x})$$

$$@_{\mathrm{x}}^{\mathrm{lvl}}\varphi = \mathrm{EX}^{\mathrm{lvl}}(\mathrm{x} \wedge \varphi)$$

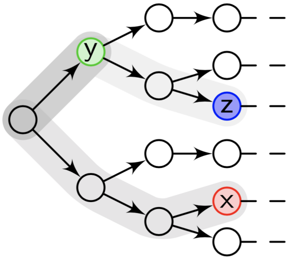


e.g. $\mathrm{nom}(\mathrm{x}, 3)$

Multiple nominals

Let $\mathrm{diff\text{-}nom}(x_1, \ldots, x_n, lvl)$ be

$$\bigwedge_{i \in [1,n]} \mathrm{nom}(x_i, k) \wedge \bigwedge_{i<j \in [1,n]} \neg @_{x_i}^{lvl} x_j$$

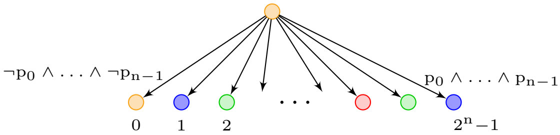


$\mathrm{nom}(y, 1) \ \wedge \ \mathrm{diff\text{-}nom}(x, z, 3)$

Enforcing exponential degree

An example of local nominals technique

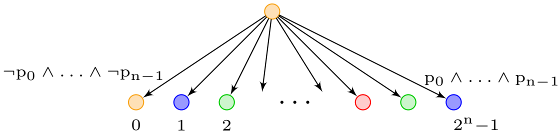
- Label children with bits $P = \{p_0, p_1, \ldots, p_{n-1}\}$.



Enforcing exponential degree

An example of local nominals technique

- Label children with bits $P = \{p_0, p_1, \ldots, p_{n-1}\}$.



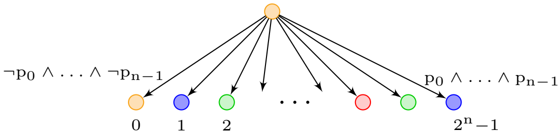
- There exists a node carrying zero.

$$\mathrm{EX}(\neg p_0 \wedge \neg p_1 \ldots \wedge \neg p_{n-1})$$

Enforcing exponential degree

An example of local nominals technique

- Label children with bits $P = \{p_0, p_1, \ldots, p_{n-1}\}$.



- There exists a node carrying zero.

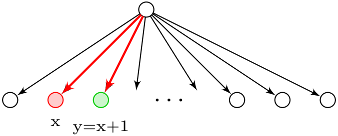
$$EX(\neg p_0 \wedge \neg p_1 \ldots \wedge \neg p_{n-1})$$

- There are no two nodes with the same number.

$$\forall x, y \text{ diff-nom}(x, y, 1) \rightarrow \neg(\bigwedge_{p \in P} @^1_x p \leftrightarrow @^1_y p)$$

Successor relation (aka. adding plus one)

$$\underbrace{\forall x \; \mathrm{nom}(x, 1) \wedge x \neq 2^{n}-1}_{\text{for all nodes } x \text{ except the last}} \rightarrow \underbrace{\exists y \; \text{diff-nom}(x, y, 1) \wedge y = x+1}_{\text{there is a successor } y}$$



How to express $y = x + 1$?

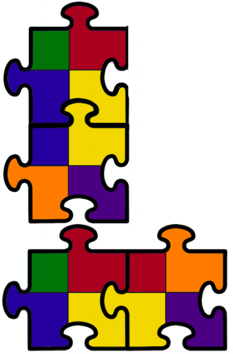
$$\begin{array}{rcccccccccccccl} & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & & x \\ + & & & & & & & & & & & & 1 & & 1 \\ \hline & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & & y = x + 1 \end{array}$$

$$\bigwedge_{i=0}^{n-1} \left(\underbrace{@_x^1 \left(\neg p_i \wedge \bigwedge_{j=0}^{i-1} p_j \right)}_{\text{look for the first zero bit}} \rightarrow \left(\underbrace{@_y^1 \left(\bigwedge_{j=0}^{i-1} \neg p_j \wedge p_i \right)}_{\text{reset previous bits, set } p_i} \wedge \underbrace{\bigwedge_{j=i+1}^{n-1} @_x^1(p_j) \leftrightarrow @_y^1(p_j)}_{\text{rewrite other bits}} \right) \right)$$

How to prove TOWER–hardness? Part I: k–Tillings

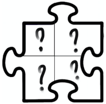
$$\exp(1, n) = 2^n, \; \exp(k+1, n) = 2^{\exp(k,n)}$$

Constraints



Rules

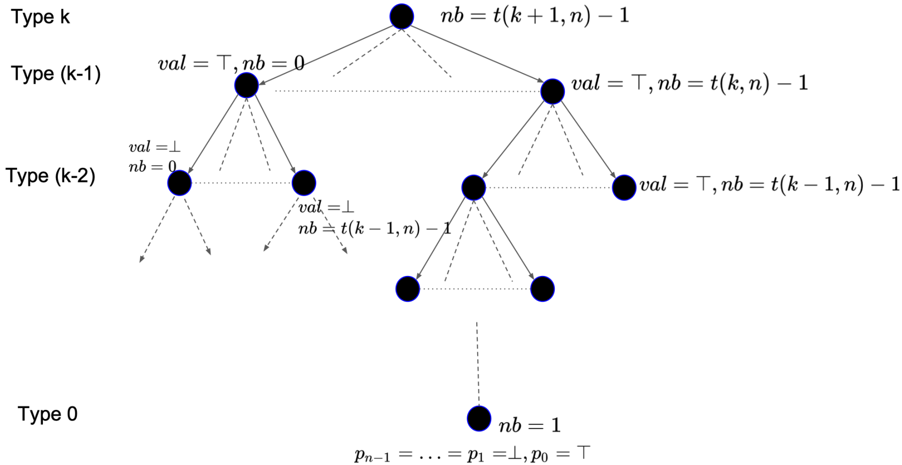
- Finite set of puzzles



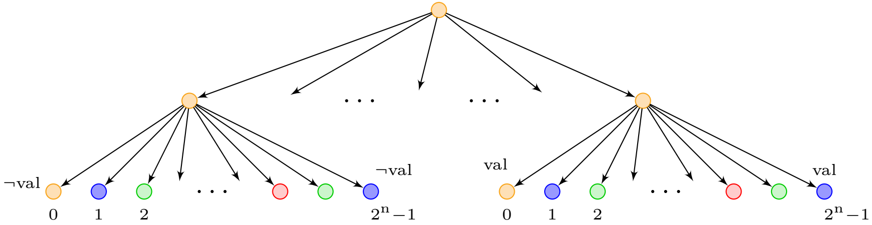
- Horizontal and vertical constraints
- Goal: Tile a board of the size

$$\exp(k, n) \times \exp(k, n)$$

How to prove TOWER-hardness? Part II: Huge degree

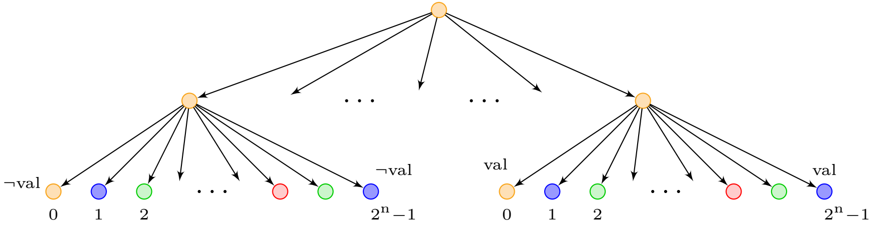


Enforcing doubly-exponential degree



Number of a node $\rightsquigarrow$ encoded on val predicates

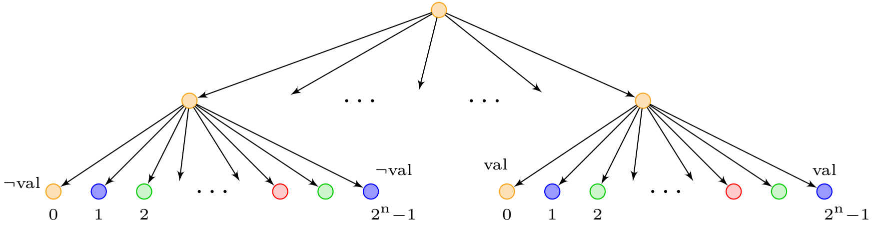
Enforcing doubly-exponential degree



Number of a node $\rightsquigarrow$ encoded on val predicates

- There exists a node carrying zero. EX(AX(¬val))

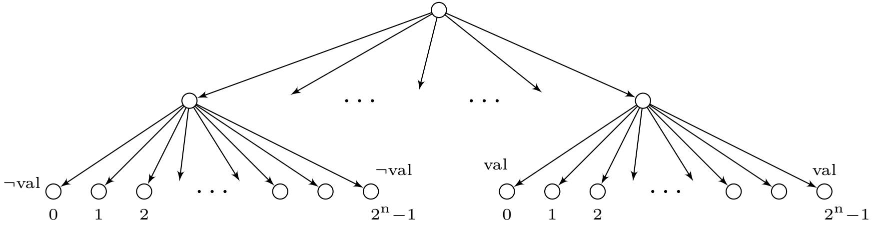
Enforcing doubly-exponential degree



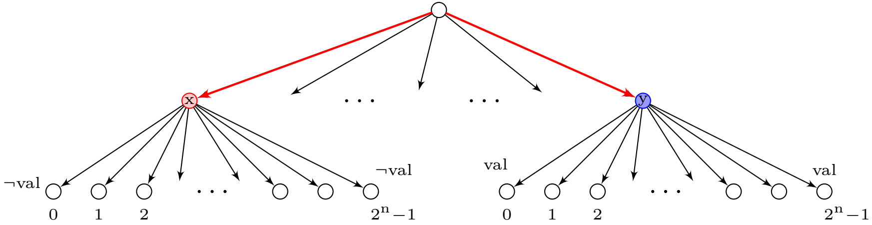
Number of a node ⇝ encoded on val predicates

- There exists a node carrying zero. EX(AX(¬val))
- There are no two nodes with the same number. ???
- Every node has successor. ???

There are no two nodes with the same number.

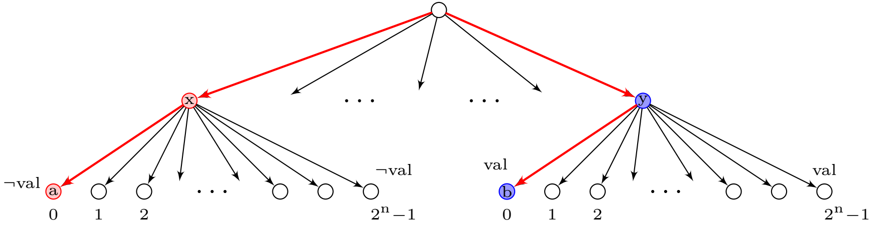


There are no two nodes with the same number.



$$\forall x, y \text{ diff-nom}(x, y, 1) \rightarrow \neg \text{ equalNum}(x, y)$$

There are no two nodes with the same number.

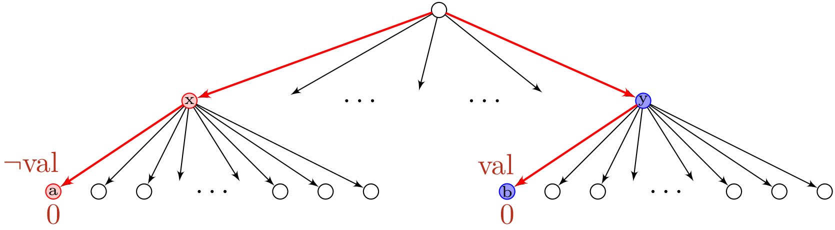


$$\neg\,\text{equalNum}(x, y) \stackrel{\text{def}}{=}$$

$$\exists a \exists b \left(\text{diff-nom}(a, b, 2) \wedge @^1_x(\text{EX } a) \wedge @^1_y(\text{EX } b)\right) \wedge$$

$$\text{equalNum}(a, b) \wedge \neg \left(@^2_a(\text{val}) \wedge @^2_b(\text{val})\right)$$

There are no two nodes with the same number.

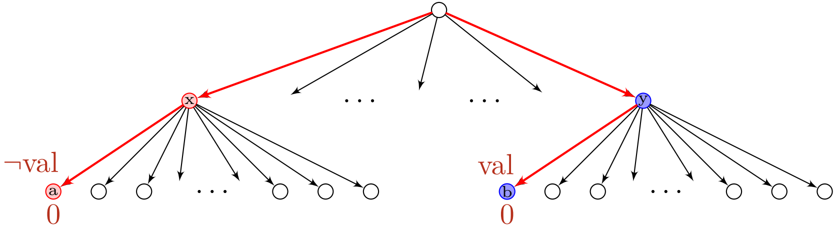


$$\neg\,\mathrm{equalNum}(x, y) \stackrel{\mathrm{def}}{=}$$

$$\exists a \exists b \left(\mathrm{diff\text{-}nom}(a, b, 2) \wedge @^1_x(\mathrm{EX}\ a) \wedge @^1_y(\mathrm{EX}\ b)\right) \wedge$$

$$\mathrm{equalNum}(a, b) \wedge \neg\left(@^2_a(\mathrm{val}) \wedge @^2_b(\mathrm{val})\right)$$

There are no two nodes with the same number.



$$\neg\,\text{equalNum}(x, y) \stackrel{\text{def}}{=}$$
$$\exists a \exists b\,\big(\text{diff-nom}(a, b, 2) \wedge @^1_x(\text{EX } a) \wedge @^1_y(\text{EX } b)\big) \wedge$$
$$\text{equalNum}(a, b) \wedge \neg\,\big(@^2_a(\text{val}) \wedge @^2_b(\text{val})\big)$$

What about successor relation?

More general way of adding plus one

an abstraction of the previous technique

$$
\begin{array}{rcccccccccccccl}
 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & & x \\
+ & & & & & & & & & & & & 1 & & 1 \\
\hline
 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & & y = x + 1
\end{array}
$$

A nice abstraction:

x	left, to be rewritten	selector $= 0$	right $= 111\ldots1$
x+1	left, to be rewritten	selector $= 1$	right $= 000\ldots0$

Summing it up

A part of the main result

$\mathrm{QCTL}_{\mathrm{EX}}$ on trees is k–NExpTime–hard for each $k \in \mathbb{N}$.

Summing it up

A part of the main result

$\mathrm{QCTL}_{\mathrm{EX}}$ on trees is k–NExpTime–hard for each $k \in \mathbb{N}$.

Reduction was uniform, so $\mathrm{QCTL}_{\mathrm{EX}}$ is TOWER–hard.
The upper bound from MSO on trees.

Summing it up

A part of the main result

$\mathrm{QCTL}_{\mathrm{EX}}$ on trees is k–NExpTime–hard for each $k \in \mathbb{N}$.

Reduction was uniform, so $\mathrm{QCTL}_{\mathrm{EX}}$ is TOWER–hard.
The upper bound from MSO on trees.

The main result

Over trees $\mathrm{QCTL}_{\mathrm{EX}}$ is TOWER–complete.

Summing it up

A part of the main result

$\mathrm{QCTL}_{\mathrm{EX}}$ on trees is k–NExpTime–hard for each $k \in \mathbb{N}$.

Reduction was uniform, so $\mathrm{QCTL}_{\mathrm{EX}}$ is TOWER–hard.
The upper bound from MSO on trees.

The main result

Over trees $\mathrm{QCTL}_{\mathrm{EX}}$ is TOWER–complete.

Main ingredient = Huge degree.
What happens when degree is bounded?

Trees with bounded degree

Bounded degree trees

$\mathrm{QCTL}_{\mathrm{EX}}$ is AExpPol-complete on trees with bounded degree.

AExpPol = alternating exp time with poly alternations

Main ingredients:

Trees with bounded degree

Bounded degree trees

$\mathrm{QCTL}_{\mathrm{EX}}$ is AExpPol-complete on trees with bounded degree.

AExpPol = alternating exp time with poly alternations

Main ingredients:

- Upper bound = exp models + model checking algorithm
- Lower bound = exp multi-tilings [Bozzelli et al, 2018]

Conclusions

Our results (arbitrary trees)

Quantified K (aka. $\mathrm{QCTL_{EX}}$) on trees is TOWER-complete.
Hardness applies also to $\mathrm{GL}, \mathrm{S4}, \mathrm{K4}, \mathrm{KD}, \mathrm{QCTL_{EF}}$ on trees.

Our results (bounded degree trees)

$\mathrm{QCTL_{EX}}$ is AExpPol-complete on trees with bounded degree.

Conclusions

Our results (arbitrary trees)

Quantified K (aka. $\mathrm{QCTL}_{\mathrm{EX}}$) on trees is TOWER-complete.
Hardness applies also to $\mathrm{GL}, \mathrm{S4}, \mathrm{K4}, \mathrm{KD}, \mathrm{QCTL}_{\mathrm{EF}}$ on trees.

Our results (bounded degree trees)

$\mathrm{QCTL}_{\mathrm{EX}}$ is AExpPol-complete on trees with bounded degree.

Open problems:

- Expressive power of $\mathrm{QCTL}_{\mathrm{EX}}$?
- Nice elementary fragments ?

Conclusions

Our results (arbitrary trees)

Quantified K (aka. $\mathrm{QCTL}_{\mathrm{EX}}$) on trees is TOWER-complete.
Hardness applies also to $\mathrm{GL}, \mathrm{S4}, \mathrm{K4}, \mathrm{KD}, \mathrm{QCTL}_{\mathrm{EF}}$ on trees.

Our results (bounded degree trees)

$\mathrm{QCTL}_{\mathrm{EX}}$ is AExpPol-complete on trees with bounded degree.

Open problems:

- Expressive power of $\mathrm{QCTL}_{\mathrm{EX}}$?
- Nice elementary fragments ?