On the Computational Complexity of Intuitionistic Modal and Description Logic

Edward Hermann Hausler
Mario Benevides
Valeria de Paiva
Alexandre Rademaker

Departamento de Informática - PUC-Rio - Brasil
Coppe-UFRJ
FGV - Brasil
Univ. Birmingham - UK

EMAP November 2011

Easy tasks versus Hard tasks

Basic considerations on *SAT*

- It is easy to verify that a boolean formula is truth under an assignment.
- Is it easy to find/build an assignment that satisfies a boolean formula ?
- What is “easy” in computational terms ??
- Worst case analysis for algorithms.
- Best algorithm analysis for problems.

Easy tasks versus Hard tasks

Finding an assignment that satisfies a boolean formula α, considering that each million assignments is evaluated in 1 sec. Naive Algorithm

k	2^k	Time
5	32	insignificant
10	1024	0.001 seg
16	65536	0.06 seg
20	1048×10^3	1 seg
32	4.29×10^9	1h 12 min

Easy tasks versus Hard tasks

Problems as hard as *SAT*

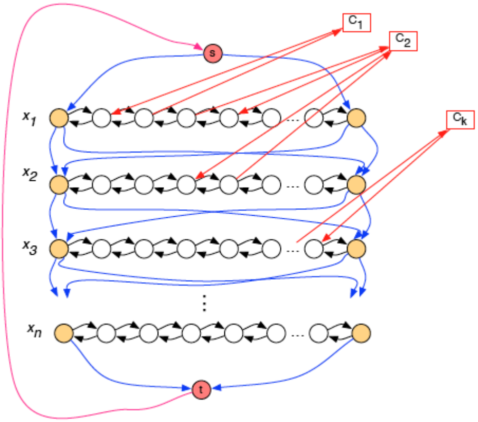
- Given a directed graph G, is there a cycle that visits every vertex exactly once? (*Hamilton*)
- Given n cities, and distances $d(i,j)$ between each pair of cities, does there exist a path of length $\leq k$ that visits each city exactly once? (*TravelingSalesman*)
- It is easy to verify that a route is a hamiltonian cycle in the graph. How about finding a route that is a hamiltonian cycle ?
- An efficient solution to *Hamilton* carries with it an efficient solution to *SAT*.
- An efficient solution to *SAT* carries with it an efficient solution to *Hamilton*.

Time used to find a cycle in a Graph with *k* vertexes

The computer verifies 1 million routes in 1 sec.

k	(k - 1)!	Cálculo total
5	24	insignificante
10	362 880	0.3 seg
15	87 bilhões	24 hs e 6 min
20	1.2×10^{17}	3 milhões de anos
25	6.2×10^{23}	0.19×10^{17} anos

Solving *SAT* using *Hamilton*



Some Complexity Classes under Cook-Karp-thesis

Time Classes

$$\mathcal{P} \subset \mathcal{NP} \subset \mathcal{EXP} \subset \mathcal{NEXP}$$

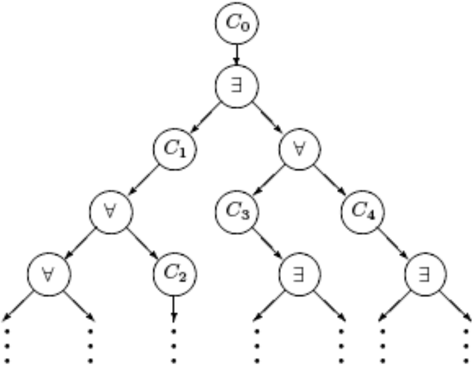
Some Complexity Classes under Cook-Karp-thesis

Space Classes

$$\mathcal{L} \subset \mathcal{NL} \subset \mathcal{PSPACE} = \mathcal{NPSPACE} \subset \mathcal{EXPSPACE} \subset \mathcal{NEXPSPACE}$$

Alternating Turing Machines

∃-accepting states and ∀-accepting states.



Facts and main uses of ATM

- $\mathcal{NP} = \mathcal{APTIME}/\exists$.
- $\mathcal{CONP} = \mathcal{APTIME}/\forall$.
- $\mathcal{APTIME} = \mathcal{PSPACE}$ and $\mathcal{APSPACE} = \mathcal{EXP}$.
- $\mathcal{APTIME}$ complete problems concerns knowing in a 2-person perfect information game, whether player 1 has a winning strategy (Games against Nature).
- *BQF*, is *APTime*-complete and hence *PSPACE*-complete.
- Intuitionistic Logic (IPL), many Modal Logics (S4, KT, K, etc) and the core of the Description Logics (ALC) are *PSPACE*-complete.

Computational Complexity of Combined Modal Logics

Curious Phenomena

- K is *PSPACE* and $K \times K$ is *PSPACE*.
- $K4$ is *PSPACE*-complete, but $K \times K4$ is *EXPTIME*-complete.
- $S5$ is *NP*-complete, $S5 \times S5$ is coNEXPTIME-completes and $S5 \times S5 \times S5$ is undecidable.
- $(<, \omega)$ and $K4$ are PSPACE-complete, but their product is undecidable.
- *Int* is *PSPACE* and *IK* (*Int* $\times$ K) is *PSPACE*-complete. (Open ??)

The logic *IK*

The language of IK is described by the following grammar.

$$A ::= P \mid \bot \mid \neg A \mid A \wedge A \mid A \vee A \mid A \rightarrow A \mid \Box A \mid \Diamond A$$

Let $\mathcal{M} = \langle W, \leq, R, V \rangle$ be a Kripke model for IK, $w \in W$ and α be an IK formula. The satisfaction relation, $\mathcal{M}, w \models \alpha$, is defined inductively as follows:

- A $\mathcal{M}, w \models P$, iff, $P \in V(w)$
- B $\mathcal{M}, w \not\models \bot$
- C $\mathcal{M}, w \models \alpha \wedge \beta$, iff, $\mathcal{M}, w \models \alpha$ and $\mathcal{M}, w \models \beta$
- D $\mathcal{M}, w \models \alpha \vee \beta$, iff, $\mathcal{M}, w \models \alpha$ or $\mathcal{M}, w \models \beta$
- NEG $\mathcal{M}, w \models \neg\alpha$, iff, for all w', $w \leq w'$, $\mathcal{M}, w' \not\models \alpha$
- IMP $\mathcal{M}, w \models \alpha \rightarrow \beta$, iff, for all w', $w \leq w'$, if $\mathcal{M}, w' \models \alpha$ then $\mathcal{M}, w' \models \beta$
- IBOX $\mathcal{M}, w \models \Box\alpha$, iff, for all w', $w \leq w'$, for all v', $w'Rv'$, $\mathcal{M}, v' \models \alpha$.
- IDIA $\mathcal{M}, w \models \Diamond\alpha$, iff, there is v, wRv, $\mathcal{M}, v \models \alpha$.

$\leq$ and R are not independent

$$
\begin{array}{ccc}
\begin{tikzcd}
w' \arrow[r, dotted, "R"] & v' \\
w \arrow[u, "\leq"] \arrow[r, "R"] \arrow[ur, phantom, "(F1)"] & v \arrow[u, dotted, "\leq"']
\end{tikzcd}
& \text{and} &
\begin{tikzcd}
w' \arrow[r, dotted, "R"] & v' \\
w \arrow[u, dotted, "\leq"] \arrow[r, "R"] \arrow[ur, phantom, "(F2)"] & v \arrow[u, "\leq"']
\end{tikzcd}
\end{array}
$$

Metatheorems on IK

- *IK* is sound and complete regarded *IK* frames.
- *IPL* $\subset$ *iALC* (hardness is PSPACE)
- Alternating Polynomial Turing-Machine to find out winner-strategy on the SAT-Game adapted from Areces2000 (upper-bound is PSPACE).

IK is *PSPACE*-complete

$SAT_{IK} \subset PSPACE$

- One wants fo verify whether $\Gamma \to \gamma$ is satisfiable.
- $\Gamma \to \gamma$ is satisfiable, if and only if, $(\sqcap_{\theta \in \Gamma} \theta) \to \gamma$ is satisfiable in a model of Γ. A game is defined on $\Gamma \cup \{\gamma\}$
- $\exists$*loise* starts by playing a list $\{L_0, \ldots, L_k\}$ of $\Gamma \cup \{\gamma\}$-Hintikka I-sets, and two relations $\mathcal{R}$ and $\preccurlyeq$ on them.
- $\exists$*loise* loses if she cannot provide the list as a pre-model.
- $\forall$*belard* chooses a set from the list and a formula inside this set.
- $\exists$*loise* has to verify/extend the (pre)-model in order to satisfy the formula.
- $\Gamma \cup \gamma$ is satisfiable, iff, $\exists$*loise* has a winning strategy.

Δ-Hintikka I-set is a maximal prime consistent set of subformulas from Δ.

Fixing a missing point in the proof

The initial move of $\exists$*loise* is:

> $\exists$*loise starts by playing a list* $\{L_0, \ldots, L_k\}$ *of* $(\Gamma \cup \{\gamma\})$*-Hintikka I-sets, and two relations* $\mathcal{R}$ *and* $\preccurlyeq$ *on them.*

The following condition has to be added:

> *k should be polynomially bounded by* $\alpha = \Gamma \to \gamma$ *length.*

Can $\exists$*loise* be happy at the first move ?

The conditions on $\{L_0, \ldots, L_k\}$ list of α-Hintikka I-sets

CF1 If $L_w \preccurlyeq L'_w$ and $L_w \mathcal{R} L_v$ then there exists L'_v, such that $L'_w \mathcal{R} L'_v$ and $L_v \preccurlyeq L'_v$.

CF2 If $L_v \preccurlyeq L'_v$ and $L_v \mathcal{R} L_w$ then there is L'_w, such that $L_w \preccurlyeq L'_w$ and $L'_v \mathcal{R} L'_w$.

Here If $\beta \in \mathcal{F}(\alpha)$, $\beta \in L_i$ and $L_i \preccurlyeq L_j$, then $\beta \in L_j$.

Form $\alpha \in L_0$ and, if $L_i = L_j$ then $i = j$

CNEG for all L_i, for all $\neg\beta \in \mathcal{F}(\alpha)$, if $L_i \preccurlyeq L_j$ and $\beta \in L_j$ then $\neg\beta \notin L_i$

AND for all L_i, for all $\beta_1 \wedge \beta_2 \in \mathcal{F}(\alpha)$, if $\beta_1 \wedge \beta_2 \in L_i$ then $\beta_k \in L_i$, $k = 1, 2$

OR for all L_i, for all $\beta_1 \vee \beta_2 \in \mathcal{F}(\alpha)$, if $\beta_1 \vee \beta_2 \in L_i$ then either $\beta_1 \in L_i$ or $\beta_2 \in L_2$

CIMP for all L_i, for all $\beta_1 \rightarrow \beta_2 \in \mathcal{F}(\alpha)$, if $L_i \preccurlyeq L_j$, $\beta_1 \in L_j$ and $\beta_2 \notin L_j$ then $\beta_1 \rightarrow \beta_2 \notin L_i$

CIDIA for all $\diamond\beta \in \mathcal{F}(\alpha)$, if $L_i \mathcal{R} L_j$ and $\diamond\beta \notin L_i$ then $\beta \notin L_j$

CIBOX for all L_i and L_j, for all $\Box\beta \in \mathcal{F}(\alpha)$, if $L_i \preccurlyeq L_j$, $L_j \mathcal{R} L_h$ and $\beta \notin L_h$, then $\Box\beta \notin L_i$

And $\forall$*belard* goes on $\{L_0, \ldots, L_k\}$...

If $\exists$*loise* does not lose when she presents $\{L_0, \ldots, L_k\}$ then the match continues and $\forall$*belard* may follow one of the two items below:

- **MODAL** $\forall$*belard* must choose three sets L_i, L_j, L_h, $L_i \preccurlyeq L_j$, $L_j \mathcal{R} L_h$ and a formula $A \in L_j$ to attack and $\exists$*loise* must respond according to the following items:
  - **DIA** If A is $\diamond\beta$, then $\exists$*loise* must provide an α-Hintikka set Y, such that: $\beta \in Y$ and for all $\diamond\gamma \in \mathcal{F}(\alpha)$, if $\diamond\gamma \notin L_j$ then $\gamma \notin Y$. For all $\Box\gamma \in \mathcal{F}(\alpha)$, if $\Box\gamma \in L_j$ then $\Box\gamma \in L_j$ and $\gamma \in Y$.
  - **BOX** If A is $\Box\beta$, then $\exists$*loise* must provide an α-Hintikka set Y, such that: $\beta \in Y$ and for all $\Box\gamma \in \mathcal{F}(\alpha)$, for each L_k, $L_k \preccurlyeq L_j$, such that $\Box\gamma \in L_k$ then $\gamma \in Y$. For all $\diamond\gamma \in \mathcal{F}(\alpha)$, if $\diamond\gamma \notin L_j$ then $\gamma \notin Y$.
  - **IntProp** For all $\neg\gamma \in \mathcal{F}(\alpha)$, if $\neg\gamma \in L_h$ then $\gamma \notin Y$. For all $\gamma_1 \rightarrow \gamma_2 \in \mathcal{F}(\alpha)$, if $\gamma_1 \rightarrow \gamma_2 \in L_h$ then either $\beta_1 \notin Y$ or $\beta_2 \in Y$.
- **INTUI** $\forall$*belard* must choose three sets L_i, L_j, L_h, $L_j \preccurlyeq L_h$, $L_i \mathcal{R} L_j$ and a formula $A \in L_i$ to attack and $\exists$*loise* must respond according to the following items:
  - **Imp** If A is $\beta_1 \rightarrow \beta_2$, then $\exists$*loise* must provide an α-Hintikka set Y, such that, either $\beta_1 \notin Y$ or $\beta_2 \in Y$. For all $\neg\gamma \in \mathcal{F}(\alpha)$, if $\neg\gamma \in L_i$ then $\gamma \notin Y$. For all $\gamma_1 \rightarrow \gamma_2 \in \mathcal{F}(\alpha)$, if $\gamma_1 \rightarrow \gamma_2 \in L_i$ then either $\beta_1 \notin Y$ or $\beta_2 \in Y$.
  - **Neg** If A is $\neg\beta$, then $\exists$*loise* must provide an α-Hintikka set Y, such that, $\beta \notin Y$. For all $\neg\gamma \in \mathcal{F}(\alpha)$, if $\neg\gamma \in L_i$ then $\gamma \notin Y$. For all $\gamma_1 \rightarrow \gamma_2 \in \mathcal{F}(\alpha)$, if $\gamma_1 \rightarrow \gamma_2 \in L_i$ then either $\beta_1 \notin Y$ or $\beta_2 \in Y$.
  - **Modal** For all $\diamond\gamma \in \mathcal{F}(\alpha)$, if $\diamond\gamma \notin Y$ then $\gamma \notin L_h$. For all $\Box\gamma \in \mathcal{F}(\alpha)$, if $\gamma \notin L_h$, then $\Box\gamma \notin Y$.
- **STOP1** In any of the items above, if the Y $\exists$*loise* provides is among the α-Hintikka sets already on the match, then the game stops and $\exists$*loise* win.
- **STOP2** If $\forall$*belard* cannot provide any of the three sets stated in items **MODAL** and **INTUI**, under the respective conditions, then $\exists$*loise* wins.

Ensuring that k is polynomially bounded by $lenght(\alpha)$

- We can see the list $\{H_0, \ldots, H_k\}$ together with the relations $\mathcal{R}$ and $\preccurlyeq$ in a tree-form.
- We prove that this tree has a polynomially bounded height regraded $lenght(\alpha)$.
- We prove that this tree has $log_2(lenght(\alpha))$ possible ramifications.
- We conclude the polynomial bound on k

Technical Lemmas

Lemma

Let $L = \{H_0, H_1, \ldots, H_k\}$ be a list of α-Hintikka sets satisfying the conditions stated in 14. Let $\{H_{p_0}, \ldots, H_{p_i}, \ldots, H_{p_n}\}$ be a maximal sub-list of L, such that for all $j = 0, n-1$, $H_{p_j} \mathcal{R} H_{p_{j+1}}$ or for all $j = 0, n-1$, $H_{p_j} \preccurlyeq H_{p_{j+1}}$, and for all $j_1, j_2 = 0, n$, if $H_{p_{j_1}} = H_{p_{j_2}}$ then $j_1 = j_2$. So, k is polynomially bounded by $l(\alpha)$.

Lemma

Let $L = \{H_0, H_1, \ldots, H_k\}$ be a list of α-Hintikka sets satisfying the conditions stated in 14. Let $\{H_{p_0}, \ldots, H_{p_i}, \ldots, H_{p_n}\}$ be a maximal sub-list of L, such that for all $j = 1, n$, there are $j_1 \neq j_2$, $j_1, j_2 = 0, k$ such that, $H_{p_j} \mathcal{R} H_{j_1}$ and $H_{p_j} \preccurlyeq H_{j_2}$. Under these conditions, $n = c.log(l(\alpha))$.

THANK YOU