

Automatická konstrukce modelů

Radek Miček



Řešená úloha

- › Hledání konečných modelů v klasické logice prvního řádu
- › Vstup:
 - množina klauzulí
 - a velikost domény.
- › Výstup:
 - jeden model,
 - nebo všechny navzájem neizomorfní modely dané velikosti.

Metoda MACE – úvod

- › Idea: Problém převedeme na SAT a k jeho vyřešení použijeme standardní SAT řešič.
- › Příklad:
 - Vstup: klauzule $f(x) = y \vee P(x)$, velikost domény 2.
 - Zavedeme výrokové proměnné pro f a P : $A_{f(0)=0}$, $A_{f(0)=1}$, $A_{f(1)=0}$, $A_{f(1)=1}$, $A_{P(0)}$, $A_{P(1)}$.
 - Přidáme klauzule, jenž zajistí, že f má v každém bodě právě jednu hodnotu.
 - Vstupní klauzuli zakódujeme do výrokových klauzulí:
 - › $A_{f(0)=0} \vee A_{P(0)}$ (pro $x = 0, y = 0$),
 - › $A_{f(0)=1} \vee A_{P(0)}$ (pro $x = 0, y = 1$),
 - › $A_{f(1)=0} \vee A_{P(1)}$ (pro $x = 1, y = 0$),
 - › $A_{f(1)=1} \vee A_{P(1)}$ (pro $x = 1, y = 1$).

Metoda MACE – zplošťování

- › Do výrokových klauzulí lze snadno zakódovat ploché klauzule.
 - Klauzule $f(x) = y \vee P(x)$ je plochá.
- › Neploché klauzule je třeba transformovat na ploché (zploštit).
 - Klauzule $f(c) = y$ není plochá.
 - Transformací získáme ekvivalentní plochou klauzuli $x \neq c \vee f(x) = y$.
- › Zplošťování může zvýšit počet proměnných v klauzuli.

Metoda MACE – vlastnosti

- › Počet výrokových proměnných závisí exponenciálně na počtu proměnných v klauzuli.
 - Příklad: Je-li velikost domény n , pak klauzule s p proměnnými bude zakódována do n^p výrokových klauzulí.
 - Důsledek: Metoda MACE není bez dalších modifikací praktická.
- › Lze použít existující SAT řešič.
- › Velmi dobré výsledky v praxi.
 - Program Paradox, implementace metody MACE s modifikacemi, vyhrál 6 ročníků soutěže CASC.

Crossbow

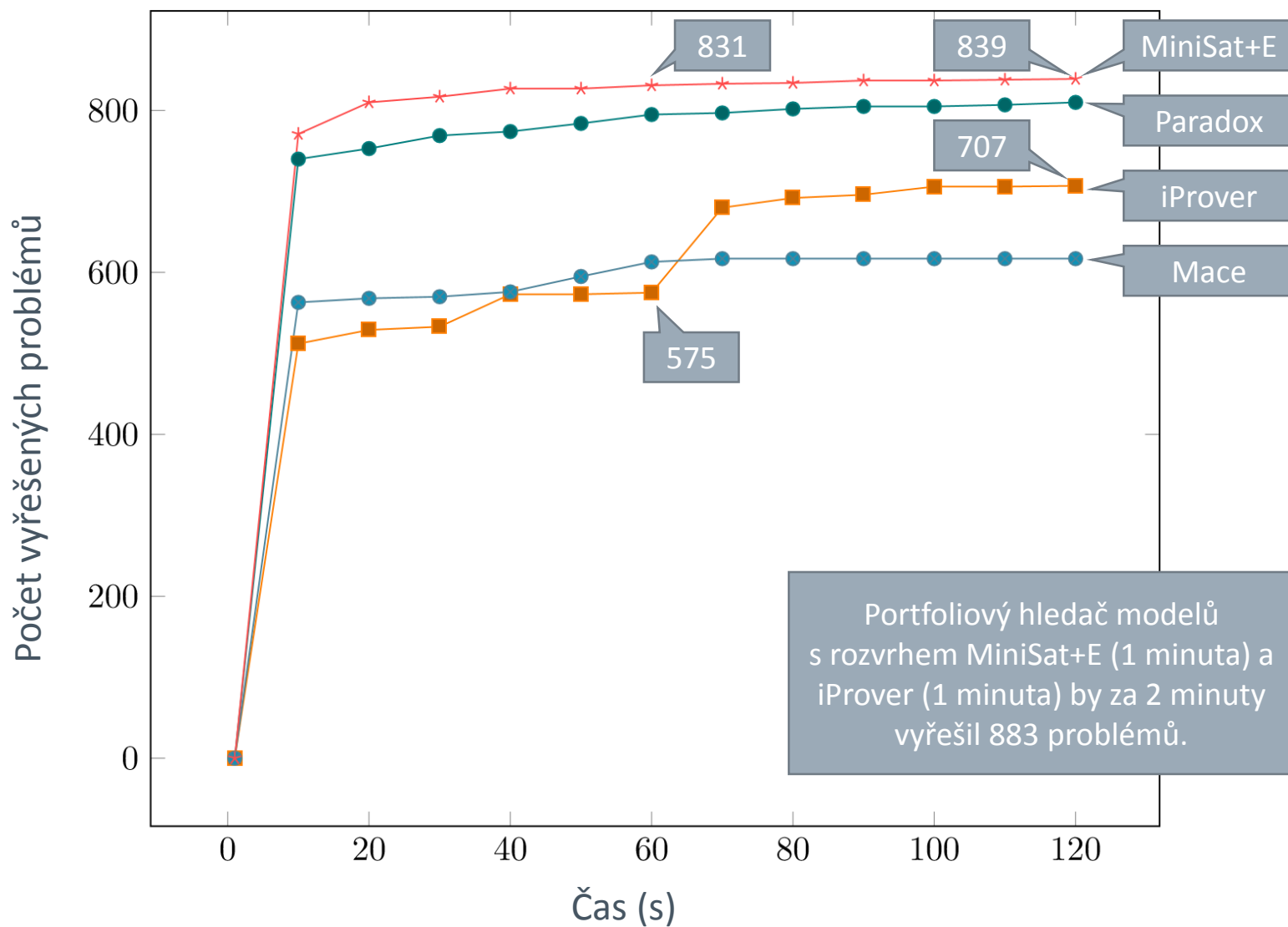
- › Program Crossbow je naše implementace metody MACE.
- › Crossbow = metoda MACE + všechny modifikace z programu Paradox + nové modifikace.
- › Nové modifikace:
 - Odzplošťování.
 - Převod do omezujících podmínek řešiče Gecode (zplošťování není nutné).
 - Využití speciálních vlastností některých teorií.
 - Generování redundantních klauzulí pomocí dokazovače E.
 - Josat – modifikovaný MiniSat pro nižší spotřebu paměti.

Experimentální srovnání

Počty vyřešených
problémů z databáze TPTP
za 2 minuty, 11 GiB paměti.

	Počet problémů	Mace	Paradox	iProver	CMSat+E	MiniSat+E	Josat+E	Gecode+E
ALG	37	14	25	1	30	30	30	1
BOO	21	14	16	10	17	17	17	15
CAT	10	8	10	10	10	10	10	10
GEO	18	0	7	4	7	7	7	1
GRP	85	72	78	71	80	80	79	77
HWV	41	12	21	14	27	28	28	19
KRS	13	8	8	8	8	8	8	8
LAT	62	62	54	13	61	61	61	50
LCL	44	23	43	23	43	43	43	25
LDA	26	24	21	0	26	25	25	0
MGT	11	10	11	11	11	11	11	11
NLP	236	158	209	230	222	223	223	180
NUM	15	5	5	5	5	5	5	5
PUZ	27	17	22	20	23	23	23	18
RNG	14	8	10	8	10	10	10	8
SET	30	5	5	5	5	5	5	5
SWW	39	4	23	11	24	24	23	14
SYN	223	142	195	221	182	182	182	137
TOP	19	8	19	19	19	19	19	8
Ostatní	44	23	28	23	28	28	28	23
Celkem	1015	617	810	707	838	839	837	615

Experimentální srovnání



Shrnutí

- › Navrhli jsme a vyzkoušeli jsme nové modifikace metody MACE.
- › Crossbow s řešičem MiniSat je obvykle rychlejší než Paradox a spotřebuje méně paměti.
- › Pro každý z testovaných programů existují problémy, pro něž je daný program lepší než ostatní testované programy.
- › Z testovaných programů pouze Crossbow a Mace4 podporují hledání všech neizomorfních modelů dané velikosti.

Děkuji za pozornost!

π