# Report - Backbones

David Surma

Úlohu řeším následujícím postupem. Nejprve spustím solver klasicky a zjistím, jestli existuje vůbec nějaký model. Pokud ne, neexistují žádné backbones, v opačném případě se kandidáti na backbones nachází v nalezeném splňujícím ohodnocení.

Poté v cyklu procházím tyto kandidáty na backbones. Pro každý z nich spustím znovu solver, kterému přidám klauzuli obsahující negaci zkoumaného literálu. Díky tomu solver hledá modely obsahující negaci zmíněného literálu. Pokud žádný takový model nenajde, potom je dle definice daný literál backbone. Jinak je nalezeno nějaké další splňující ohodnocení původní formule a kandidáti na backbones se zúží na průnik tohoto ohodnocení a původních kandidátů.

Úlohu jsem otestoval na 10 instancích splnitelných problémů obsahujících 50 proměnných a 218 klauzulí. Výsledky zachycuje tabulka 1. Vzhledem k použitému algoritmu platí, že počet běhů solveru musí být větší než počet backbones. V několika případech došlo k tomu, že počet běhů solveru byl maximální (51), v tom případě však bylo backbones taky velké množství (alespoň 44). V ostatních případech algoritmus počet běhů solveru ořezal, průměrně byl asi o 10 větší než počet skutečných backbones. Výjimkou byla instance uf50-05.cnf, kde se nacházaly pouze 4 backbones a na jejich nalezení bylo potřeba 20 běhů.

Tyto rozdíly záleží hlavně na tom, jaké další splňující ohodnocení solver najde, což je víceméně náhodné, případně by mohlo jít heuristicky vylepšit za použití nějakých assumptions.

| | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| backbones | 44 | 47 | 28 | 45 | 4 | 48 | 22 | 49 | 37 | 36 |
| solver runs | 51 | 51 | 39 | 50 | 20 | 51 | 28 | 51 | 46 | 43 |

Tabulka 1: Výsledky hledání backbones u problémů uf50-XX.cnf