V posledních letech automatická formální verifikace software pokročila z několika výzkumných laboratoří do rozsáhlých aplikací, jako je cloudová infrastruktura a smart kontrakty. Formální verifikační techniky založené na technice model checking poskytují nezbytné záruky *úplným* a *automatickým* zkoumáním chování systémů. Navíc poskytují *svědky* (vysvětlení) pro výsledek své analýzy: chybové chování, pokud nějaké existuje, nebo důkaz o absenci takového chování.

Obecný problém, který se automatická verifikace software snaží vyřešit, je však nerozhodnutelný. Navzdory této teoretické překážce je v mnoha případech, které se v praxi vyskytují, docela efektivní. Tento (možná překvapivý) úspěch připisujeme kombinaci faktorů: neutuchajícímu úsilí výzkumníků, kteří přicházejí s novými verifikačními postupy pro třídy problémů, kde existující techniky narážejí na své možnosti; úžasný pokrok v základních technologiích řešení splnitelnosti, zvláště v *Satisfiability Modulo Theories* (SMT); a zvýšení dostupného výpočetního výkonu prostřednictvím paralelních a cloudových výpočtů. Nicméně rostoucí složitost systémů v reálném světě představuje nové výzvy pro formální verifikace, zejména pokud jde o škálovatelnost.

Úloha automatické verifikace software má dvě části: modelování úlohy ve formálním rámci a řešení výsledného matematického problému. I když je modelování netriviálním krokem v procesu verifikace, bylo široce řešeno a existuje mnoho konceptů modelování vhodných pro různé systémy. Na druhé straně, hledání řešení výsledného problému je úzkým hrdlem, pokud jde o složité moderní programy. Tato práce se zaměřuje na tuto druhou část problému, kde se hledají nové efektivní přístupy. Předpokládáme, že problémy jsou modelovány *symbolicky*, formulemi v logice prvního řádu. Konkrétně pracujeme v logickém rámci *Hornových klauzulí s omezujícími podmínkami* (constrained Horn clauses – CHC) a zkoumáme matematický problém *rozhodování splnitelnosti* CHC systému. Splnitelnost CHC zobecňuje úkol ověřování *safety vlastností* v *přechodových systémech*, což je model rozšířený ve formálním verifikaci. Tento úkol je složitý a obecně nerozhodnutelný již tehdy, pokud jazyk omezujích podmínek obsahuje lineární celočíselnou aritmetiku. V naší práci tvrdíme, že k tomuto úkolu lze přistoupit na různých úrovních, které označujeme jako vrstvy *základů*, *verifikačných algoritmů* a *kooperace*. Ty odpovídají rozhodovacím a interpolačním procedurám, sekvenčním algoritmům model checkingu a přístupům s využitím multiagentních systémů. Dále tvrdíme, že další (vyšší) vrstvy nejen navazují na předchozí (nižší) vrstvy, ale s nimi také významně interagují. Práce na vyšších vrstvách může významně těžit z hlubokého pochopení vrstev pod nimi. Celkově posouváme oblast automatické verifikace software novými řešeními na všech třech vrstvách.

Na základní vrstvě přispíváme novým interpolačním algoritmem pro konflikty v teorii lineární aritmetiky. Tento algoritmus rozšiřuje standardní přístup založený na Farkasově lemmatu a počítá logicky silnější interpolanty. Experimentální vyhodnocení aplikace pro model checking ukazuje, že s naším interpolačním algoritmem může stejný algoritmus model checkingu úspěšně vyřešit některé problémy, na kterých diverguje s původním interpolačním algoritmem.

Na verifikační vrstvě vynalézáme koncept sekvence *přechodové mocninné abstrakce* (transition power abstraction – TPA) a přinášíme algoritmy pro model checking založené na TPA, které řeší známý problém detekce hlubokých protipříkladů v přechodových systémech. Navíc sekvence TPA může být zdrojem kandidátů na *přechodový* invariant. To umožňuje algoritmům založeným na TPA prokázat bezpečnost systémů prostředky, které jsou do značné míry ortogonální ke stávajícím technikám.

Na podporu vývoje verifikačních technik jsme vyvinuli nástroj Golem, nový řešič splnitelnosti CHC systémů. Hlavními rysy toho řešiče jsou jeho těsná integrace s interpolačním SMT řešičem a podpora pro více back-endových řešících algoritmů. Golem má sloužit především jako výzkumný nástroj pro další zkoumání algoritmů pro model checking založených na SMT a algoritmů pro obecnou splnitelnost Hornových klauzulí s omezujícími podmínkami. Golem byl klíčem pro rychlý vývoj naší prototypové implementace algoritmů založených na TPA. Je však také efektivní ve srovnání s jinými CHC řešiči v poslední edici mezinárodní soutěže CHC-COMP. Golem tedy

může sloužit jako efektivní back-end pro doménově specifické nástroje, které modelují různé verifikační úlohy v rámci CHC. Golem je již možné použít jako back-end pro Korn, nástroj na verifikaci programů v jazyce C.

Na kooperativní vrstvě přispíváme abstraktním rámcem, který zobecňuje koncepty z algoritmů pro model checking založených na indukci. Abstrakce se explicitně zaměřuje na aplikaci ve scénáři řešení s více agenty, kde si více instancí stejného řešiče vyměňuje informace a tímto způsobem spolupracuje na řešení jediné instance problému. Tento abstraktní rámec používáme k odvození paralelní verze úspěšného algoritmu PD-KIND a pomocí experimentů ukazujeme, že výměna informací může výrazně zlepšit výkon. Protože PD-KIND spoléhá na interpolaci jako podproceduru, používáme náš nový interpolační algoritmus k získání rozmanitějšího chování agentů, což přispívá velkou měrou ke zlepšení výkonu.