

Dynamické funkce programovacích jazyků, jako je dynamický typový systém, dynamické volání funkcí, dynamické vykonávání kódu a dynamické datové struktury, poskytují flexibilitu, která urychluje vývoj. Tyto funkce ale snižují množství informací, které jsou kontrolovány v době kompilace. To má za následek nižší výkon a větší chybovost programů. Tento problém je možné vyřešit pomocí technik statické analýzy. Dynamické funkce bohužel pro tyto techniky představují překážku a zásadně omezují jejich přesnost, spolehlivost a výkonnost. Abychom tento problém pomohli vyřešit, navrhujeme framework pro statickou analýzu, který automaticky řeší dynamické funkce, a tím umožňuje definovat přesné a spolehlivé statické analýzy podobně jako v případě, kdy program dynamické funkce neobsahuje. Aby bylo takový framework možné vytvořit, navrhujeme novou techniku heap analýzy, která modeluje asociativní pole a (prototypové) objekty. Dále navrhujeme analýzu hodnot proměnných, která zjišťuje další informace potřebné pro vypořádání se s dynamickými funkcemi. Nakonec navrhujeme techniku, která umožňuje automaticky a genericky kombinovat analýzu hodnot proměnných s heap analýzou.