Populární objektově-orientované programovací jazyky jako Java nebo Scala typicky dovolují měnit objekty přiřazením nových hodnot do jejich datových položek. Zároveň je však běžné psát kód, který k objektům přistupuje pouze pro čtení. Technika Reference mutability (měnitelnost referencí) umožňuje omezit změny objektů tím, že pomocí typů rozlišuje reference na ty, které jsou pouze pro čtení, a ty, které umožňují objekt měnit. Pro jazyk Java byla tato technika již podrobně rozvinuta a existuje implementace ve formě rozšíření pro kompilátor. Scala je programovací jazyk, který se stále rozvíjí a do svého typového systému integruje mnoho pokročilých vlastností, z nichž nejpříznačnější jsou typy závislé na cestě (path-dependent types). Protože komplexnost těchto vlastností vzbuzuje obavy o možnost vzniku porušení typové bezpečnosti (type soundness), existuje formální kalkulus Dependent Object Types (DOT), který formálně dokazuje bezpečnost základních vlastností typového systému jazyka Scala.

Tato práce se věnuje možnosti využití vlastností DOT kalkulu pro integraci systému reference mutability. Přinášíme definici kalkulu roDOT, který je založen na variantě DOT kalkulu s proměnlivými objekty, a který reprezentuje měnitelnost reference pomocí speciálního vnořeného typu (type member). Toto kódování umožňuje využít typů závislých na cestě, takže je možné se z typu reference odkázat na měnitelnost jiné reference. Využívá také typů průniku a sjednocení (intersection and union types) k implementaci viewpoint adaptation (měnitelnost reference se přizbůsobuje pohledu přes jinou referenci). Kromě nutného přizpůsobení důkazu typové bezpečnosti našemu rozšíření, definujeme a dokazujeme Garanci neměnnosti (Immutability guarantee), jež formálně zaručuje, že objekty mohou být měněny pouze prostřednictvím měnitelných referencí.

Moderně psaný kód často též využívá čisté metody, které jsou naimplementovány tak, že se chovají jako matematické funkce. Rozšíření ReIm pro jazyk Java ukázalo spojitost měnitelnosti referencí s čistotou metod, konkrétně s jednou vlastností svázanou s čistotou – absenci vedlejších efektů. V této práci se věnujeme podmínkám čistoty metod v roDOT kalkulu, a formulujeme Garanci absence vedleších efektů (SEF guarantee), která zaručuje, že metody, jejichž parametry jsou neměnitelné reference, nemají vedlejší efekty. Implementace myšlenek systému ReIm do roDOT kalkulu vyžadovalo menší množství změn v typovém systému, ale kvůli nim bylo nutné přepracovat velkou část důkazu typové bezpečnosti. Garanci absence vedleších efektů jsme dokázali s použitím naší Garance neměnnosti.

Jako další vlastnost kterou je roDOT kalkulus schopen posktynout, formulujeme Garanci transformace programu (Transformation guarantee), která zaručuje, že volání metod bez vedlejších efektů lze v programu bezpečně prohazovat, aniž by se změnil výstup programu. Tuto garanci definujeme v obecném systému pro definici transformací programů pro roDOT kalkulus, a dokazujeme s využitím Garance absence vedleších efektů.

Definici roDOT kalkulu jsme mechanizovali pro systém automatické kontroly definic a důkazů Coq, a to včetně důkazů typové bezpečnosti a všech výše uvedených garancí. Jako ukázka možnosti použití myšlenek roDOT kalkulu pro jazyk Scala, součástí práce je patch pro kompilátor Dotty, ve kterém je implementována jednoduchá kontrola měnitelnosti referencí.