C# is a strongly typed language utilizing type inference to save type annotations written by a programmer. However, the type inference is not as strong as other programming languages like Rust or Haskell. This thesis aims to propose the C# language improvement which would improve the current type inference and would be likely accepted by the C# language design team. For this achievement, we analyzed Rust's type inference and observed necessary language requirements and type inference restrictions based on Hindley-Millner's formalization of type inference. These observations were used to propose a language improvement consisting of two parts. The first part is a C# specification change describing the improvement by adjusting formal C# specification. This part was presented to the language design team which resulted in a positive reaction where the team decided to keep moving forward with the proposal to make it available in the future C# language version. The second part is an implementation of the improvement in the official C# language compiler, Roslyn. The implementation is tested by using the original compiler tests and new tests testing the proposal functionality.