

1.5 Type Conversion within Assignment

In practice, there would be many different types of variables and constants, so the compiler must either reject certain mixed-type operations or generate appropriate coercion (type conversion) instructions.

This issue is not in the grammar that is being presented in Cooke's class, but is common in procedural languages.

Figure 8.19 shows the algorithm of Type Conversion for simple addition expressions. The concept deals with converting arguments into compatible types. Obviously, there is not always a common type between two arguments. This issue is critical as data types are allowed to become more complex.

The example presented by Ullman is relatively simple and present in most commonly used procedural and object oriented languages.