Unparse

C-Präprozessor

C-Präprozessor ist ein Tool, das den Quellcode vor dem Kompilieren manipuliert~\cite{ABKS13}. Dieses Tool bietet Möglichkeiten zur bedingte Kompilierung, zur Dateieinbindung und zur Erstellung lexikalische Makros~\cite{ABKS13}. Eine C-Präprozessor-Direktive beginnt mit # und geht bis zum ersten Whihespace-Zeichen weiter, optional kann nach der Direktive Argument im Rest der Zeile stehen. Der C-Präprozessor hat solche Anweisungen wie, #include zum Einbinden von Datein um zum Beispiel Header-Dateien wiederzuverwenden. Mit den Anweisungen #if, #else, #elif, #ifdef, #ifndef, und #endif wird die bedingte Kompilierung erzeugt. Dabei funktionieren #if, #else, #elif, und #endif vergleichbar mit dem was man aus Programmiersprachen und Pseudocode gewohnt ist. #ifdef ist ähnlich zu #if, wird aber nur dann True wenn der drauf folgender Makros definiert. #ifndef ist die Negation von #ifdef. Die Makros werden durch die Anweisung #define erstellt. Der Präprozessor ersetzt dann wehrend seiner Arbeit, dem Makronamen durch seine Definition. Während dieser Arbeit kann ein Makros definiert, umdefiniert und undefiniert werden. Der C-Präprozessor hat noch weitere Anweisungen, auf die wir nicht weiter eingehen. Der C-Präprozessor kann in anderen Programmiersprachen verwendet werden, wenn diese Sprachen syntaktisch ähnlich zu C sind. Beispiel für solchen Sprachen sind C++, Assemblersprachen, Fortran und Java. Der Grund dafür ist, dass der C-Präprozessor ist unabhängig von der zugrundeliegenden Programmiersprache ist. Eine so ähnliche Vorverarbeitungsmöglichkeit ist in vielen anderen Programmiersprachumgebungen.

. Aus diesen Grund kann

Wie so ein C-Preprozessorannotierter Code aussehen kann, ist in der Abbildung 1 Stelle \numcircledtikz{v} zu sehen(Abb.1 St.\numcircledtikz{v}).

Umsetzung der Variabilität mithilfe des C-Preprozessors