

# Kurzdarstellung Kimwitu++

( Seminarvortrag von Konrad Voigt und Glenn Schütze )

## Einleitung:

Der Name Kimwitu kommt aus dem Swaheli. „Witu-“ bedeutet übersetzt der Baum, „m“ wird für den Plural benutzt und „Ki“ adjektiviert das Wort. Also bedeutet Kimwitu bäumelig. Das „++“ stellt den Bezug zur Sprache C++ dar, da man Kimwitu++ als Erweiterung zu C++ sieht.

Mit Kimwitu++ kann man abstrakte Syntaxbäume definieren (ein abstrakter Syntaxbaum (AST) ist ein komprimierter Parsebaum, bei dem fast alle Knoten für Nichtterminalsymbole entfernt wurden). Dabei heißt bei Kimwitu++ ein Knoten Phylum (*Pl.:* Phyla). Jedes Phylum besitzt Operatoren, die die Ausprägung der Kinderknoten bestimmen.

## Eingabedatei(en):

Die Eingabedateien für Kimwitu++ enden auf .k. Es ist möglich, den gesamten Code in eine Eingabedatei zu schreiben, jedoch bietet es sich an, diese nach Sinn und Inhalt auf mehrere Dateien aufzuteilen. Kimwitu++ generiert aus der abstrakten Grammatik den AST.

Zu flex und bison wird die Schnittstelle mit Hilfe von YYSTYPE hergestellt. Mit dem Kimwitu++-Schalter `-yystype` generiert Kimwitu++ die Datei `yystype.h`. Sie enthält die benötigten Typen( für jedes Phylum: `yt_phylumname`). Durch die semantischen Aktionen in bison ist die Knotenerzeugung über den Phylumoperatorenaufruf möglich.

Die Kimwitu++-Dateien enthalten C++-Quelltext, die Phylumdefinitionen, eventuelle Regeln( `unparse` und `rewrite`) und Kimwitu++-Deklarationen.

## Unparse:

Mit Kimwitu++ kann man den erzeugten Baum auch abschreiten( Treewalking ). Dabei ist es möglich, an Termen bei gegebener Regel, die angegebenen Aktionen auszuführen. Dazu muss man lediglich die Muster definieren. Z.Bsp passt folgendes Muster auf jedes Add, welches zwei Argumente bekommt: `Add(*,*)`.

## Rewrite:

Mit Rewireregeln ist es möglich, den Baum entsprechend angegebener Regeln umzuformen. Dabei werden wieder Muster angegeben und natürlich der neue, einzusetzende Term auch.

## Links zu weiteren Informationen:

Hauptseite zu Kimwitu++:

<http://site.informatik.hu-berlin.de/kimwitu++/>

Vortragseite:

<http://www.informatik.hu-berlin.de/~schuetze>

Seminarseite:

<http://informatik.hu-berlin.de/~kunert/lehre/compilergeneratoren/>