0.1 Phasen eines Compilers?

Phasen werden in Analyse- und Synthesephasen unterteilt, nicht verzahnte Phasen werden als Pass bezeichnet.

Lexikalische Analyse Teilt die Zeichenfolgen des Quellprogramms in Grundsymbole ein, die als (Token,Attribut)-Paare dargestellt werden. Das Token ist das Symbol, das Attribut die Ausprägung. Die Lexikalische Analyse beschränkt sich auf endliche Automaten, z.B. Regular Expresions.

Syntax-Analyse Erzueugt aus den Token (nicht Attribute) der Lexikalischen Analyse einen Syntaxbaum, der der (in der Regel kontextfreien) Grammatik der Quellsprache entspricht. Arbeitet nach dem Prinzip eines Kellerautomaten. Die Blätter des generierten abstrakten Syntaxbaums (Operatorbaums) sind die Token (Terminale), die Knoten bilden die Operatoren (Nonterminale).

Semantische Analyse Behandelt die Bedeutung von Symbolen (die Beziehung der Symbole untereinander), und trägt diese in eine Symboltabelle ein. Weiters findet eie Typ-Überprüfung und Operator-Identifizierung statt (insbesondere bei überladenen Operatoren). Die Semantische-Analyse liefert einen attributierten Syntaxbaum, der das Programm (syntaktisch und statisch-symantisch) repräsentiert. Es werden Kontextbeziehungen mitteles Attributierter Grammatik behandelt.

Zwischencode-Erzeugung Linearisiert einen attributierten Syntaxbaum, insbesondere mittels Sprungbefehle, Typenkonversionen und temporären Hilfsvariablen für Zwischenergebnisse.

Code-Optimierung Es wird zischen Maschinenabhängige und unabhängier Optimierung unterschieden. Es soll das Programm vereinfacht werden, umd Laufzeit oder Speicher einzusparen. Für die Maschinenabhängige Optimierung werden spezielle Maschinenbefehle genutzt, für die Maschinenunabhängige z.B. Indexberechnungen in Schleifen vereinfacht, oder konstante Ausdrücke ausgewertet.

Code-Erzeugung Es wird ein Objektprogramm zum Ablauf auf einer realen Maschine erzeugt, z.B. durch die Ausgabe von Assemblercode, der übersetzt wird.

0.2 Was ist Front-End und Back-End eines Compilers?

Alle Phasen die unabhängig vom Zielsystem sind werden als Front-End bezeichnet, im Detail Lexikalische, Syntax und Semantische Analyse sowie Zwischencode-Erzeugung. Die Phasen das vom Zielsystem abhängigen Backends sind Code-Optimierung und Code-Erzeugung.

0.3 Was ist ein abstrakter Syntaxbaum?

Ein abstrakter Syntaxbaum (oder Operatorbaum) ist das Ergebnis der Syntax-Analyse. In den Blättern stehen Token bzw. Terminale, die Knoten sind Operatoren.

0.4 Was ist ein Kellerautomat

Ein Kellerautmat ist ein um einem Kellerspeicher erweiterter endlicher Automat. Kann z.B. $a^n b^n$ erkennen.

0.5 Was sind Kontextbeziehungen

Kontextbeziehungen entstehen, wenn die zugrundeliegenden Grammatik nicht kontextfrei ist. Z.B. müssen Bezeichner for der Benutzung deklariert werden, oder Operatoren in einem bestimmten Kontext verträglich mit den Ausdrücken und Zuweisungen sein.

0.6 Was ist ein Binder

Ein Binder (linker) fügt einzelne Programmodule zu einem ausführbaren Programm zusammen. Dabei werden Refferenzen aufgelöst und u.U. Module (Libraries) eingebunden. Dannach kann das Programm vom Loader durchgeführt werden.

0.7 Was sind die Vorteile eines Mehrpass Compilers

Die einzelne Pässe haben klare, explizite Schnittstellen. Es wird weniger Hauptspeicher benötigt (wegen der Hintereinanderausführung). Nachteilig ist die zeitaufwendige Generierung von Zwischendarstellungen.

0.8 Wie stellt man Strukturen (Arrays) im Speicher dar?

Bei Strukturen liegen die einzelnen Felder ausgerichtet hintereinander im Speicher. Entspricht die Länge eines Felders nicht der Adresslänge entsteht ein Zwischenraum (Padding). Der Zugriff erfolgt mittels Startadesse + Index*Grösse.

0.9 Was ist der Unterschied zwischen Machienencode und Assemblercode?

Maschinencode sind die Bitmuster, die Befehle repräsentieren. Assemblercode ist eine lesbare Repräsentation dieser Befehle in Mnemonics.

0.10 Was sind Aufrufkonventionen

Die Aufrufkonventionen einer Plattform legen fest, welche Register von einer aufgerufenen Funktion zerstört werden dürfen, und welche unverändert bleiben müssen. Weiters wie Argumente und Rückgabewerte übergeben werden.

0.11 Was sind caller/callee saved Register

Caller-Saved Register dürfen von einer aufgerufenen Funktion überschrieben werden, (müssen gegebenfalles von der Aufrufenden Funktion zuvor gesichert werden), Calee-Saved Register müssen von der aufgerufenen Funktion vor eine Rücksprung wiederhergestellt werden (also unveränder bleiben).

0.12 Was sind Grundsymbole

Keywords (if, while, ...), Identifier (Objektnamen für z.B. Variablen), Literals (Werte, z.B. 3.14), Delimiter (z.B. =,+,==).

0.13 Was ist ein regulärer Ausdruck

Ein regulärere Ausdruck dient zur Spezifikation des Aufbaus von Zeichen. Ein RegEx ist gleichmächtig wie endliche Automaten.

0.14 Ausgabe Aktionen

In der Analyse-Phase müssen nebst der prüfemden Analyse von Symbolen auch Umcodierugen und deren Ausgabe vorgenommen werden. Aktionen der Analyse sind:

- Zurückliefern des Token als Integer durch benannte Konstante oder Aufzählungstyp.
- Tabellenverwaltung: Bezeichner und Konstanten werden in Tabellen eingetragen, oder existierende Einträge gesucht. Ein Index-Verweis auf diesen Eintrag wird als Attribut geliefert.
- Wortsymbole (Keywords): Wird ein Keyword erkannt, so muss (einem dem Keyword eindeutig zugeordneter) Integer retouniert.
- Ausblenden von Leerzeichen und KommentarKommentaren..

0.15 Was ist eine kontextfreie Grammatik

Eine kontexfreie Grammatik ist ein Quintupel (T,N,P,S) bestehend aus den Terminalen T, Nonterminalen N, Produktionen P und einem Startsymbol S. Auf den linken Seiten der Produktionen darf nur genau ein Nonterminal stehen.

0.16 Was ist ein Wort

Ein Wort ist eine beliebige Verkettung von Nonterminalen und Terminalen.

0.17 Was ist eine (Links/Rechts-) Ableitung (Reduktion)

Eine Ableitung ist die Anwendung einer Produktion auf ein Wort, bei der ein Nonterminal des Wortes, welches linke Seite einer Produktion ist, durch die rechte Seite der Produktion ersetzt wird.

Bei der Linksableitung wird das erste Nonterminal von links, bei der Rechtsableitung von Rechts, abgeleitet.

Die Reduktion ist die Umkehrung der Ableitung.

0.18 Was ist ein Ableitungsbaum (zum Unterschied eines Syntaxbaumes)

Ein Ableitungsbaum (parse tree) ist eine Baum-Abbildung einer Ableitung. Die Innenknoten sind Nonterminale, die Blätter Terminale, die Knonten entstehen durch die Anwendung von Produktionen.

0.19 Was sind Satz, Satzform und Grammatik?

Eine Satzform ist ein Wort, welches vom Startsymbol S aus abkeitbar ist. Ein Satz besteht nur noch aus Terminalen. Die von der Grammatik G erzeugte Sprache L(G) ist die Menge all ihrer Sätze.

0.20 Wann sind Grammatiken äquivalent

Wenn sie die gleiche Sprache beschreiben.

0.21 Was sind Regular right Part Grammars (RRPG)

Eine Darstellung einer Sprache durch RegEx, wobei im Gegensatz zu Regukären Definitionen Rekusionen erlaubt sind.

0.22 Was ist First und Follow

First gibt an, mit welchen Terminalsymbolen die ableitbaren Satzformen eines Wortes einer Grammatik beginne können.

Follow gibt an, weclhe Terminalsymbole einem Nonterminal N in beliebiger Satzform folgen können.

Beide Mengen könne iterativ berechnet werden.

0.23 Shift-Reduce Analayse (auch Bottom-Up-Analyse)

Ausgehend von einem Eingabewort werden so lange Linksreduktionen durchgeführt, bis das Startsymbol erreicht wurde, oder eine weiter Ableitung nicht mehr möglich ist. Dabei wird immer der Knoten nach seine Nachfolgern erzeugt. Vorteil ist, dass die Grammatik auch Linksreduktionen und gleiche Anfänge haben darf.

0.24 Tabellen der LR Analyse

In in einer LR Analyse werden eine Action-Tabelle und eine Tabelle mit Folgezustände (Goto-Tabelle).

Die Aktionenen der Action-Tabelle sind shift (Eingabesymbol in der Keller schieben), reduce p (reduzieren mit der Produktion nummer p), accept, error.

0.25 Was ist ein zulässiger Präfix

Reduktionsansätze sind Symbolfolgen oben im Keller, die der rechten Seite einer Produktion entsprechen, und deren Ersetzung eine Schritt der entsprechenden Rechtsableitung darstellt.

Eine Folge von Anfangssymbolen einer Satzform, die rechts nicht über den Ansatz hinausgeht, heist zulässiger Präfix.

0.26 Ist LR(1) mächtiger als LR(2)

Nein. LR ist ein Analyse von links nach rechts, also eine Linksreduktion. Eine Eigenschaft von LR ist, dass die Vorrausschau 1 gleich mächtig ist, wie eine Vorrausschau größer 1.

0.27 Was macht man bei Mehrdeutikeiten

Bei mehrdeutigen Grammatiken entstehen in der LR-Analyse Mehrfach-Einträge in der Action-Tabelle, und es kann zu redcue/reduce bzw. schift/reduce Konflikten kommen. Bei erstem ist nicht klar welche Produktion reduziert werden solle, bei zweiterem ob shift oder reduce ausgeführt werden soll. Es muss eine aktion gestrichen werden, yacc kann dies automatisieren, dann wird bei shift/reduce shift, bei reduce/reduce der erste reduce ausgeführt.

0.28 Hierachie der Analyse-Verfahren

Für die Verfahren LR (Left Right), LALR (Look Ahead LR) und SLR (Simple-LR) gilt das SLR(1) eine Teilmenge der LALR(1) Grammatiken, und LALR(1) eine Teilmenge der LR(1) Grammatiken verarbeiten kann.

Die weniger mächtigen erzeugen u.U. Doppeleinträge, bei mehrdeutigen Grammatiken erzeuugt aber jedes LR Verfahren Mehrfacheinträge in der Actiontabelle. Die Menge der Kontextfreien Grammatiken (die die mehrdeutigen beinhaltet) ist mächtiger als LR.

0.29 Was ist eine attributierte Grammatik

Eine attributierte Grammatik ist eine erweiterte kontextfreie Grammatik. Jedem Grammatiksymbol können synthetisierte oder ererbte Attribute, sowie Regeln zugewiesen werden.

0.30 synthetisierte/ererbte Attribute

Ein syntetisiertes Attribut eines Knoten hängt nur von seinen Kindern ab. Ein ererbtes Attribut wird von einem Grammatiksymbol der Rechten Seite weiter gegeben, hägt also von den Vätern oder Brüdern ab.

0.31 Abhängigkeitsgraph (bei der Attributauswertung)

Durch die Zuweisungsregeln der Atribute kommt es zu Abhängikeiten der Attributer der Knoten. Diese werden im Abhängigkeitsgraphen abgebildet. Die Auswertung durch Attribut-Auswerter kann Pass- oder Visitorientiert erfolgen. Bei ersterm wird der Ableitungsbaum mehrmals durchlaufen, und bei jedem Durchlauf alle bereits auflösbaren Attribute eingetragen. Bei zweiterem wird der Ableitungsbaumentspechend des Abhängigkeitsgraphen durchlaufen.

0.32 Symboltabelle

In der Symboltabelle werden die Namen aller im Quellprogramm deklarierter Objekte, sowie deren Typ gesammelt.

0.33 Was bedeutet Befehlsauswahl, Befehlsanordnung, Registerbelegung

- Befehlsauswahl (instruction selection, code selection): Die Operationen der Zwischendarstellung werden zu Befehlen der Zilemaschine zusammengefasst.
- Befehlsanordnung (instruduction scheduling): Die Befehle werden im Hinblick auf eine schnelle Ausführung angeordnet.
- Registerbelegung (register allocation): Die während früherer Phasen verwendeten Pseudoregister werden durch Maschienenregister ersetzt.

0.34 Was ist ein Datenflussgraph

Ein Datenflußgraph ist ein gerichteter Graph, dessen Knoten Instruktionen, und dessen Kanten zwischen den Knoten die zugrunde liegenden Datenanhängigkeiten beschreiben.

0.35 Was ist eine Baumgrammatik

Eine Baumgrammatik ist ein Regelsatz, mit welchem eine Zwischencodebaum in einen Maschiencodebaum gewandelt werden kann.

0.36 Duerfen Baumgrammatiken mehrdeutig sein

Ja, durch die Hinzunahme von Kosten kann dadurch der Weg mit der optimalen Ausführungszeit ermittelt werden.

0.37 Was sind Kettenregeln

Kettenregeln sind Regelen in der Form NonterminalA_>NonterminalB. Dabei müssten u.U. noch nicht berechnete Kosten von Teilbäumen berücksichtigt werden. Ein Lösung besteht darin Kettenregelen erst am Schluss zu berücksichtigen, und dadurch entstehende Abhängigkeiten wiederholt zu berechen.

0.38 Ist die Befehlsauswahl fuer Baueme (DAGs) optimal

Für Bäume kann die optimale Befehlsauswahl ermittelt werden, nicht jedoch für DAGs (azyklische Graphen).