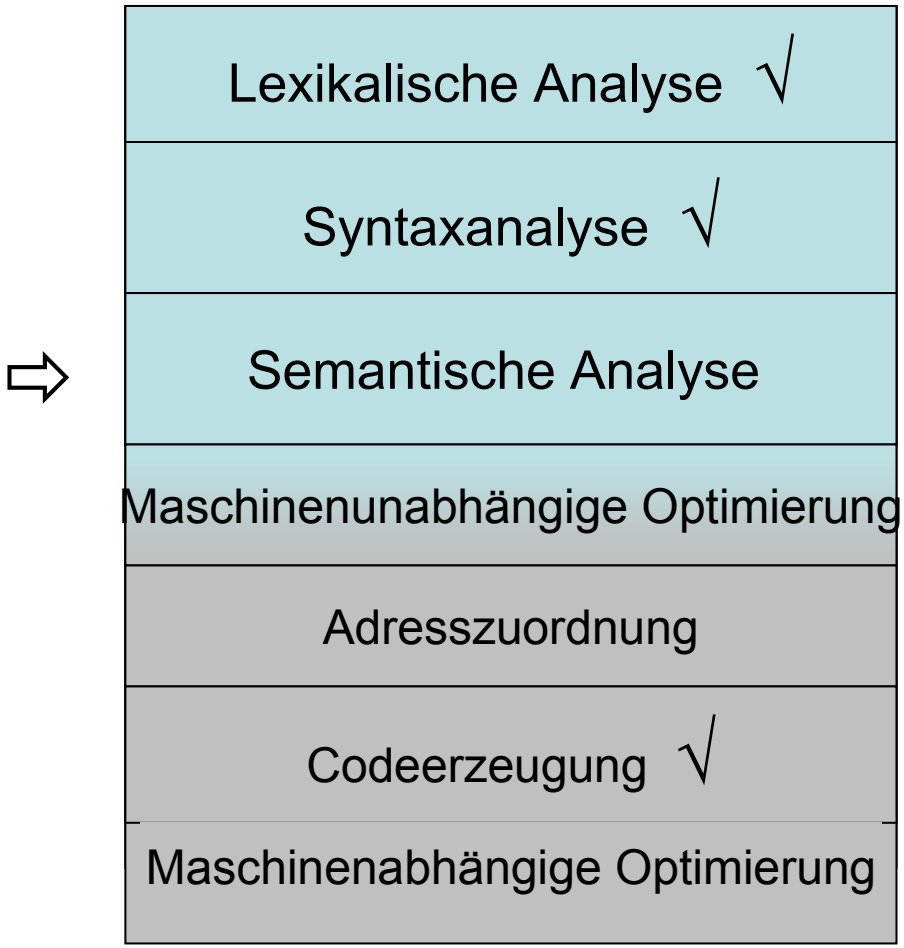
6. Die semantische Analyse

Zwischenstand:



6. Die semantische Analyse

Aufgaben:

- Deklariertheitseigenschaften
- Typkonsistenz

abhängig von

- Gültigkeitsregeln
- Sichtbarkeitsregeln

6. Die semantische Analyse

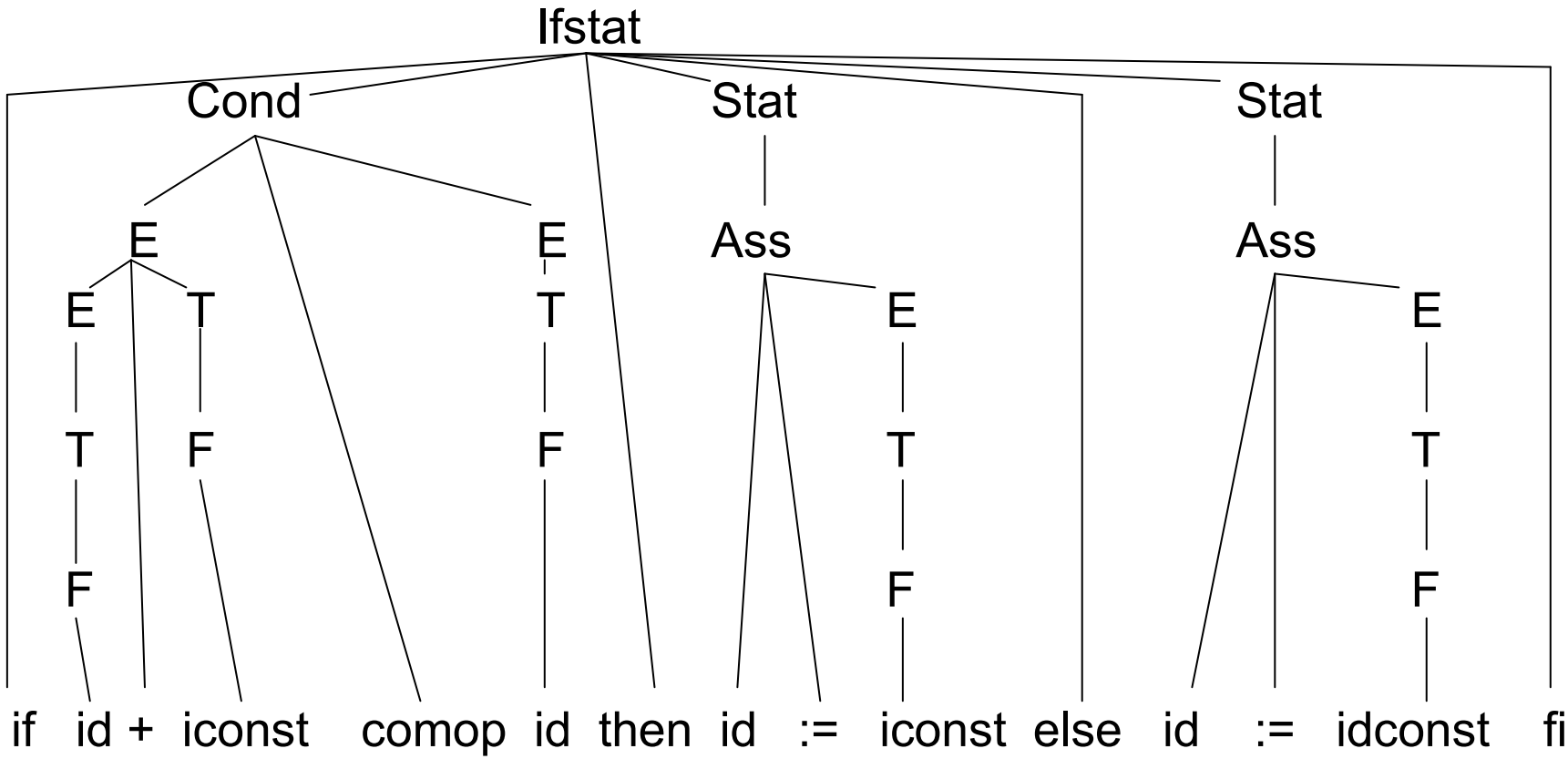
konkreter / abstrakter Syntaxbaum:

- Input der sem. Analyse: Syntaxbaum
- bisher: „konkreter“ Syntaxbaum gemäß Grammatik
- mit überflüssigen Informationen (z.B. viele Nichtterminalsymbole)
- ausreichend für sem. Analyse: abstrakter Syntaxbaum
- enthält nur wesentl. Teil der syntaktischen Struktur:
- Konstrukte und Schachtelungsbeziehung

6. Die semantische Analyse

konkreter / abstrakter **Syntaxbaum am Beispiel:**

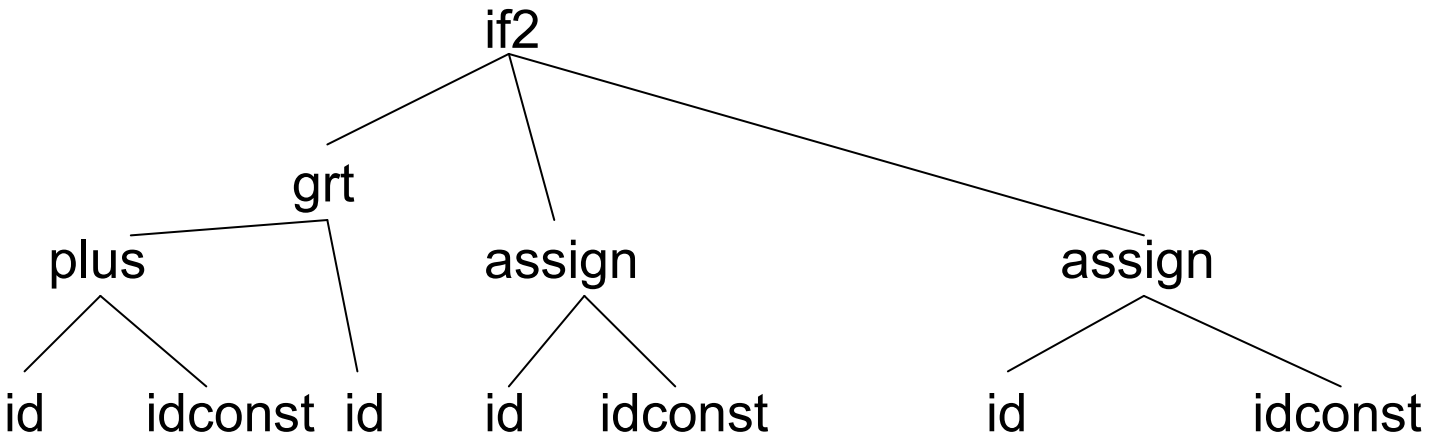
if x+1 > y **then** z := 1 **else** z := 2 **fi**



6. Die semantische Analyse

konkreter / **abstrakter Syntaxbaum am Beispiel:**

if x+1 > y **then** z := 1 **else** z := 2 **fi**



Compiler verwenden immer abstrakte Syntax zur sem. Analyse und Codeerzeugung!

6. Die semantische Analyse

Statische semantische Eigenschaften:

- bezogen auf Konstrukte der Programmiersprache
- alle (dynamischen) Ausführungen (jedes einzelnen Vorkommens) des Konstrukts haben gleichen „Wert“
- „Wert“ kann in korrektem Programm berechnet werden
- ⇨ „Wert“ der Eigenschaften können zur Übersetzungszeit berechnet werden

Beispiel: Typ eines arithmetischen Ausdrucks in typisierten Sprachen (Gegenbeispiel ?)

Dynamische semantische Eigenschaften: Wert erst zur Laufzeit bekannt

6. Die semantische Analyse

- in Quellsprache semantische Konventionen festgelegt
- statische Überprüfung (=sematische Analyse) zur Kontrolle
- Beispiele:
  1. Operator mit inkompatiblem Operanden
  2. Kontrollflussprüfung
  3. Deklariertheit von Bezeichnern
  4. Case-Label unterschiedlich
  5. Elemente in Aufzählungstyp unterschiedlich
  6. Namensüberprüfung