Teil X

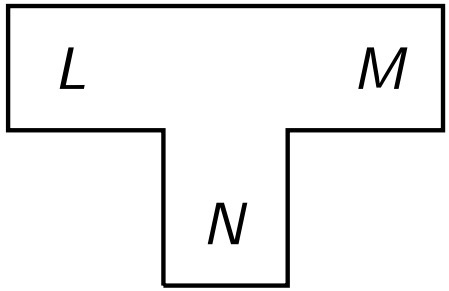
Bootstrapping

Entwicklung neuer Programmiersprachen

- Neue Programmiersprache: L
- Compiler für L:
  - in L selbst geschrieben
  - als Test für Leistungsfähigkeit (oder Mängel) der Sprache L
  - Testprogramm für korrekte Funktion des Compilers

Darstellung eines Compilers

- **T-Diagramm**:

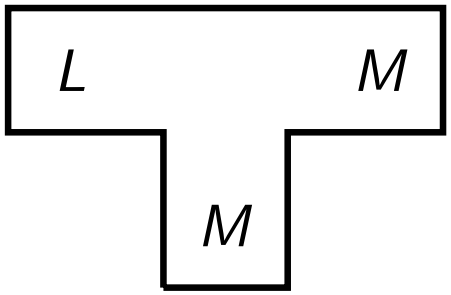


mit:

- L: Sprache, die der Compiler übersetzt (Eingabesprache)
- M: Sprache, in die der Compiler übersetzt (Ausgabesprache), oft Assembler- oder Maschinensprache
- N: Sprache, in der der Compiler geschrieben ist (Implementierungssprache)

Bootstrapping: Ziel

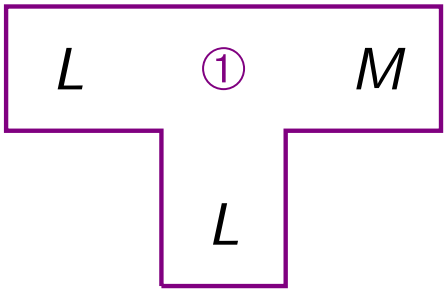
- Ziel:
  Compiler übersetzt Programme der Sprache L,
  in Programme in die (Maschinen-)Sprache M
  und läuft auf Maschine M.



- Der Compiler soll schnell laufen und geringen Platzbedarf haben.
- Der vom Compiler erzeugte Code soll schnell laufen und geringen Platzbedarf haben.

Schritt 1: vollständiger Compiler in eigener Sprache

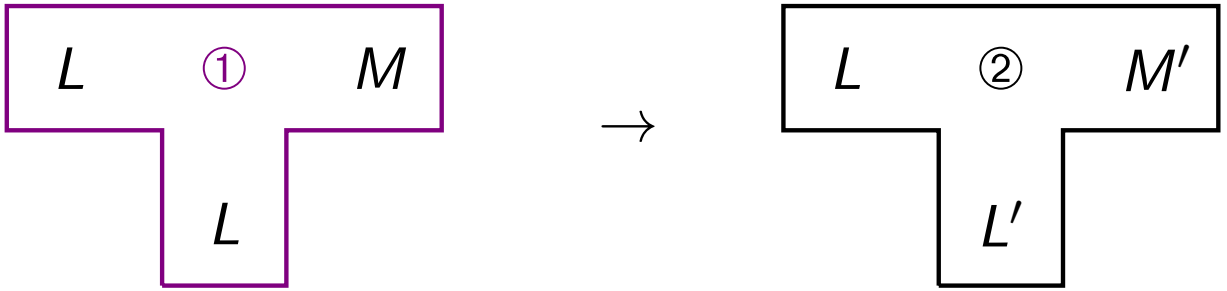
- Manuell einen Compiler erstellen für die Sprache *L* geschrieben in der Sprache *L*.



- Der Compiler soll Code erzeugen, der schnell läuft und geringen Platzbedarf hat.
- Dieser Compiler nützt uns unmittelbar nichts, da wir ja noch keine Maschine haben, die *L* versteht.

Schritt 2: Sprachumfang bei Quell- und Zielsprache abspecken

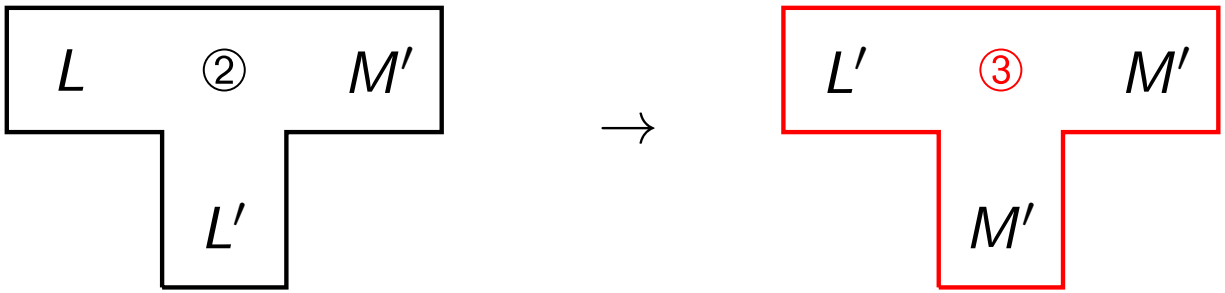
- $L' \subset L$: nur die notwendigsten Sprachkonstrukte
- $M' \subset M$: effiziente Maschinenbefehle bleiben evtl. ungenutzt
- Manuell wird aus Compiler ① ein „äquivalenter“ Compiler ② erzeugt:



- Compiler ② darf Programme erzeugen,
  die langsam laufen und großen Speicherplatzbedarf haben

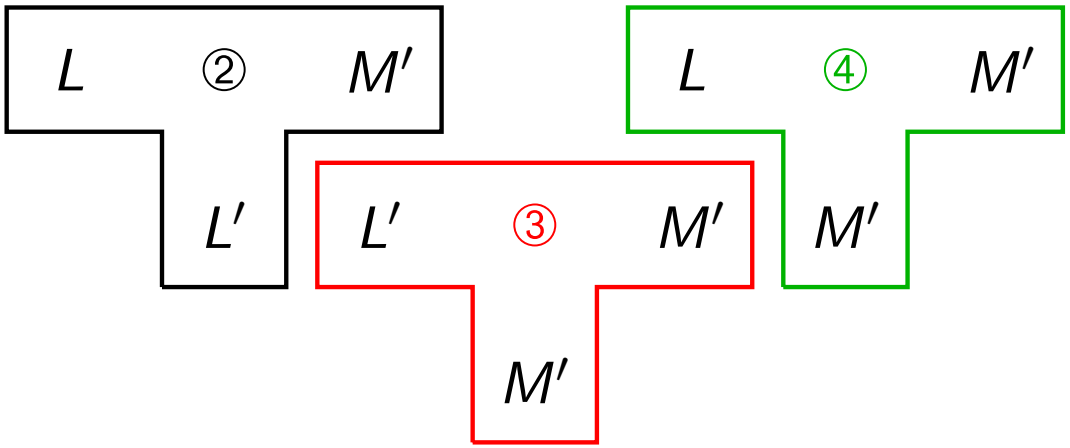
Schritt 3: Compiler von Hand nach M' übersetzen

- Letzter manueller Schritt:
  Erzeugen eines Compilers ③, der auf Maschine M' läuft,
  aus Compiler ②, etwa durch „Übersetzung per Hand“ von L' in M',
  allerdings werden nur Sprachkonstrukte aus L' übersetzt



- Compiler ③ darf langsam laufende Programme mit großem Speicherplatzbedarf erzeugen

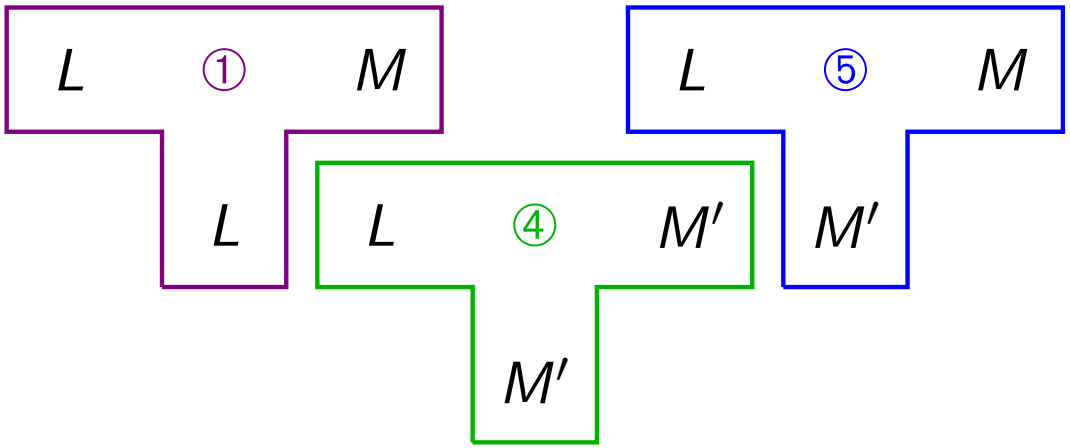
Schritt 4: Automatische Übersetzung erzeugt Compiler für Gesamtsprache L



Compiler ④ übersetzt die gesamte Sprache L nach M':

- erzeugt Programme in M', die langsam laufen und hohen Speicherplatzbedarf besitzen
- Compiler selbst kann groß sein und langsam arbeiten

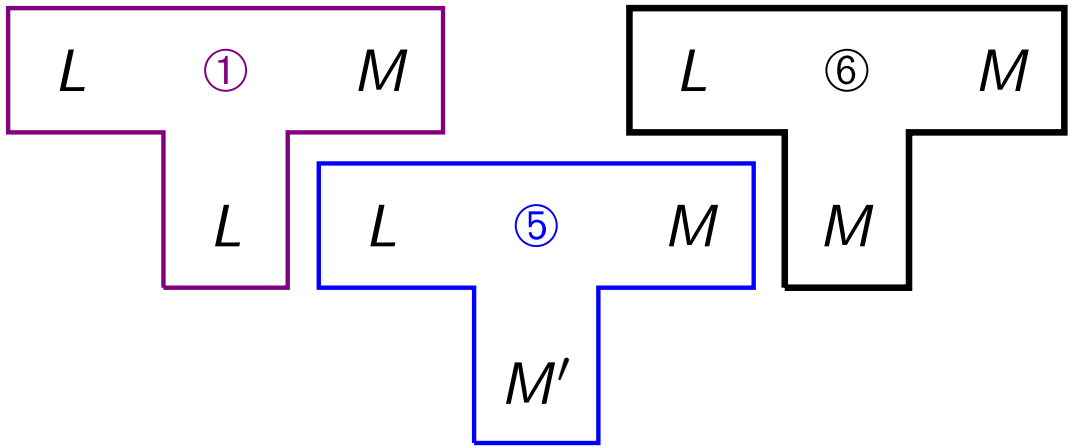
Schritt 5: 2. automatische Übersetzung erzeugt Compiler, der effizienten Code erzeugt



Compiler ⑤ übersetzt L nach M,
nutzt also den vollen Befehlssatz im übersetzten Programm:

- erzeugter Code läuft schnell und hat geringen Platzbedarf
- Compiler selbst kann groß sein und langsam arbeiten

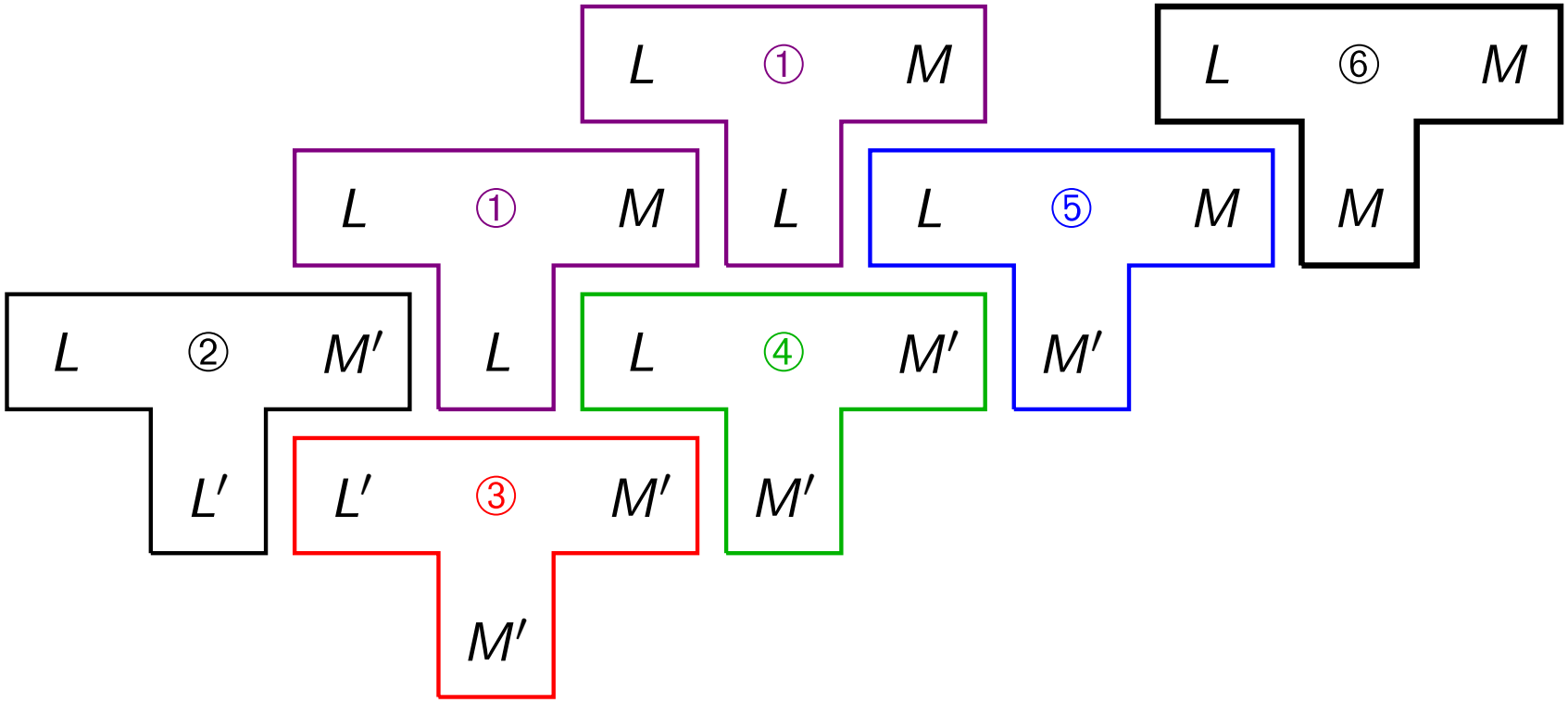
Schritt 6:
3. automatische Übersetzung erzeugt effizienten Compiler



Compiler ⑥ übersetzt L nach M
und nutzt für sich selbst den vollen Befehlssatz:

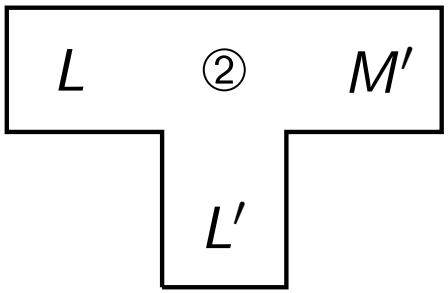
- erzeugter Code läuft schnell und hat geringen Platzbedarf
- der Compiler selbst arbeitet schnell und hat geringen Platzbedarf
  (da Compiler ⑤ derartigen Code erzeugt)

Bootstrapping der Sprache *L*

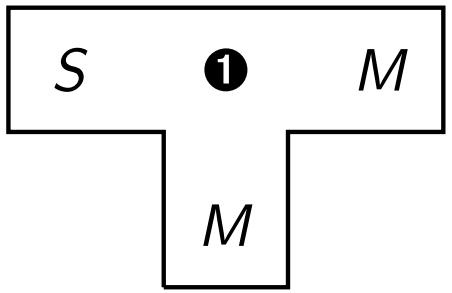


Variante zu Schritt 3: Nutzung eines vorhandenen Compilers für die Sprache S

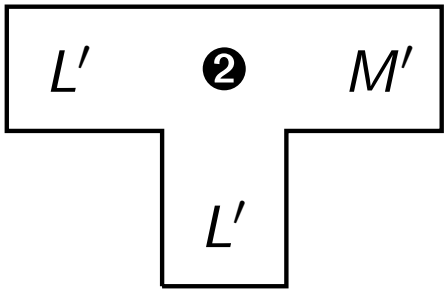
- Gegeben: u.a. Compiler ❶ für Programmiersprache S



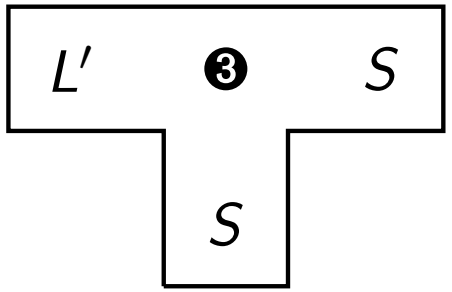
und



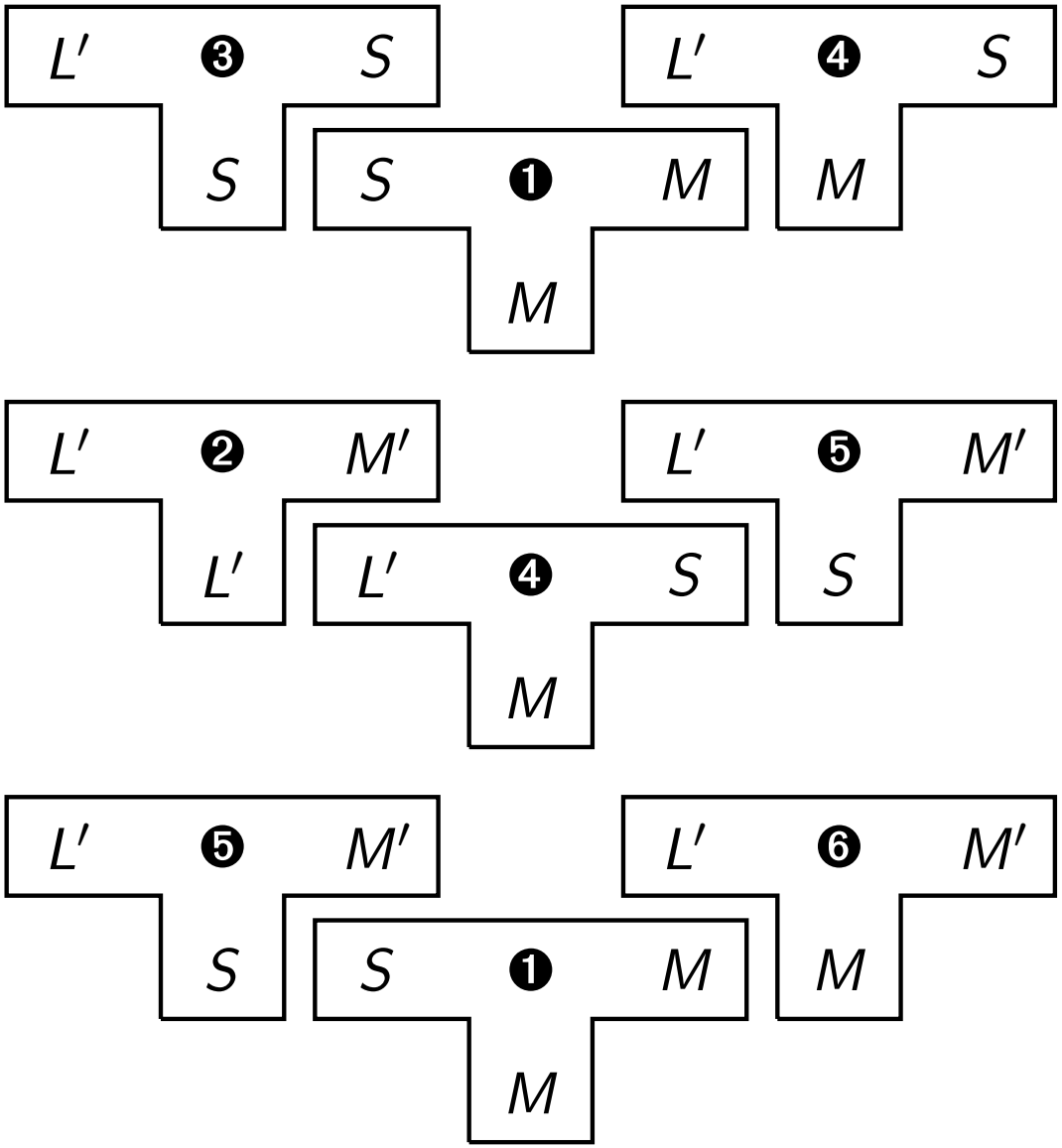
- Manuell:



und



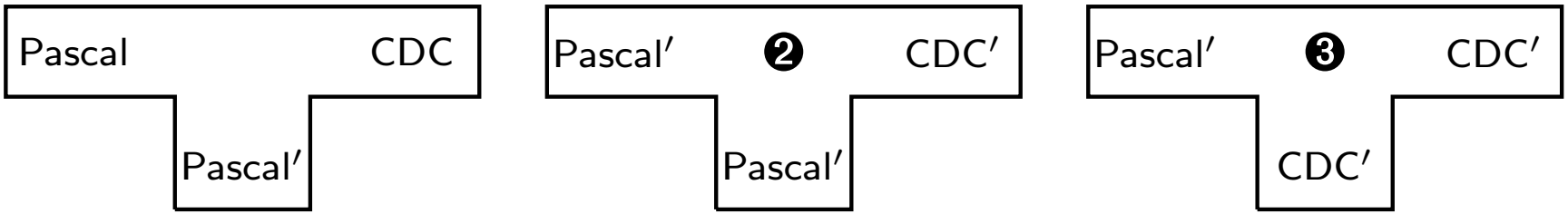
Variante zu Schritt 3



Beispiel – Erzeugung eines Pascal-Compilers

N. Wirth - Erzeugung eines Pascal-Compilers für CDC-6000

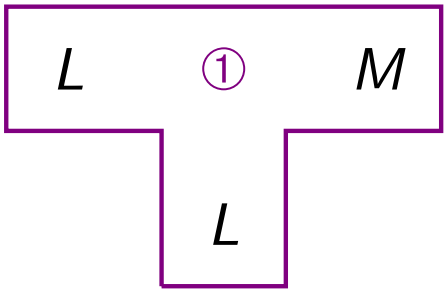
- 1968: Pascal-Compiler für CDC-6000 in FORTRAN
  - Versuch gescheitert
- 1969: Entwicklung von drei Compilern:



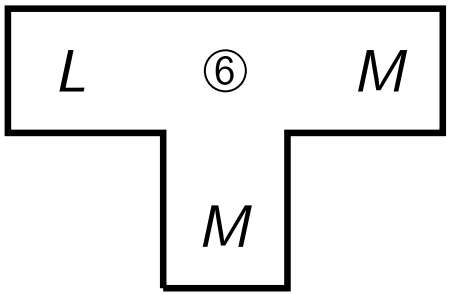
- Pascal': u.a. Verzicht auf
  - Funktionen,
  - Parameter von Prozeduren,
  - Datentypen `real`, `set`, `packed record`, `packed array`
- CDC': u.a. Verzicht auf Operationen zur
  - Gleitpunktarithmetik,
  - bitweisen Manipulationen von Wörtern (für `set` und `packed`)

Portierung eines Compilers auf Maschine N: Ziel

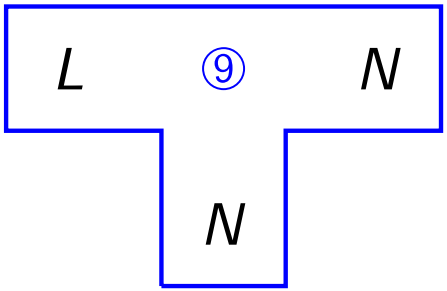
- Gegeben:



und



- Gesucht:

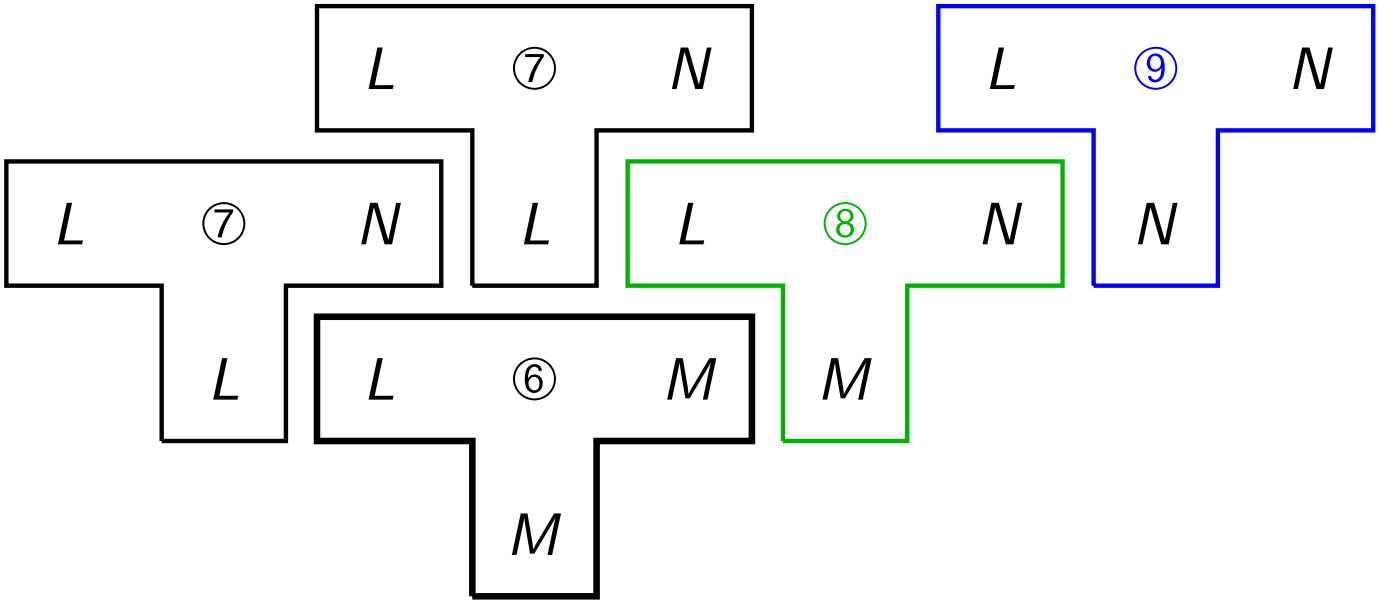


Portierung eines Compilers

Manuell aus Compiler ① den Compiler ⑦ erzeugen

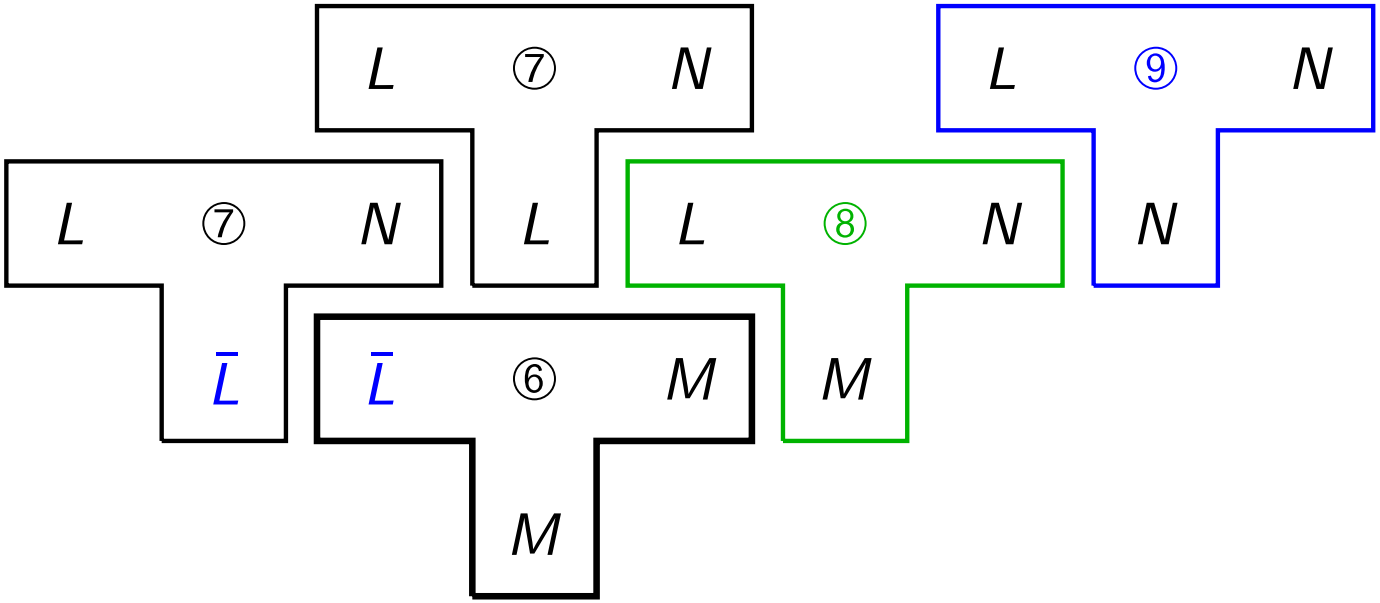
- also vor allem Code-Erzeugung und Code-Optimierung ändern

Dann Bootstrapping:



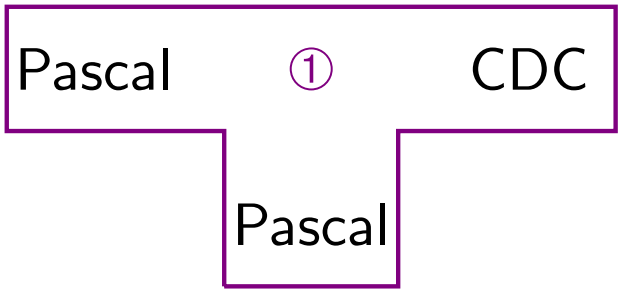
Der Compiler ⑧ läuft auf der Maschine M, erzeugt aber Programme für die Maschine N: **Cross-Compiler**.

Portierung eines Compilers - Alternative

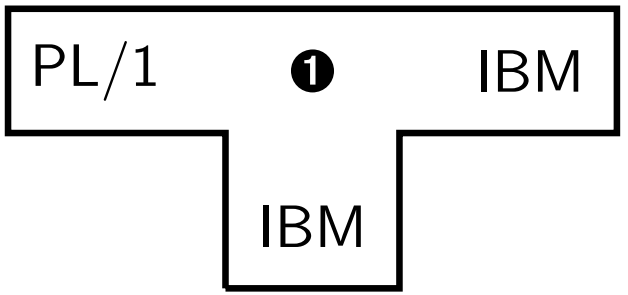


Portierung eines Compilers - Beispiel

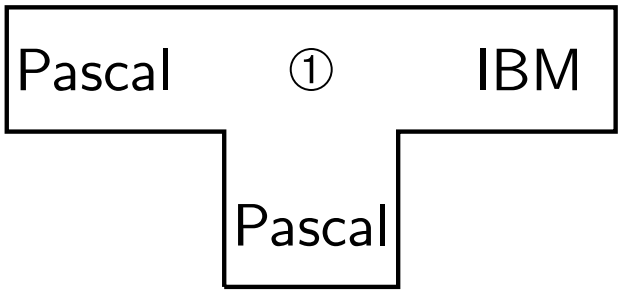
- Portierung eines Pascal-Compilers für eine CDC-6000 auf einen IBM/360 Rechner unter Verwendung eines PL/1-Compilers für die IBM/360.
- Gegeben:



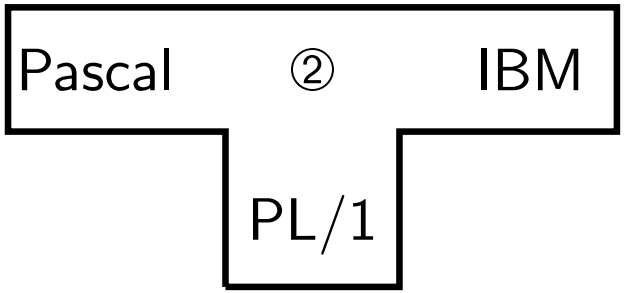
und



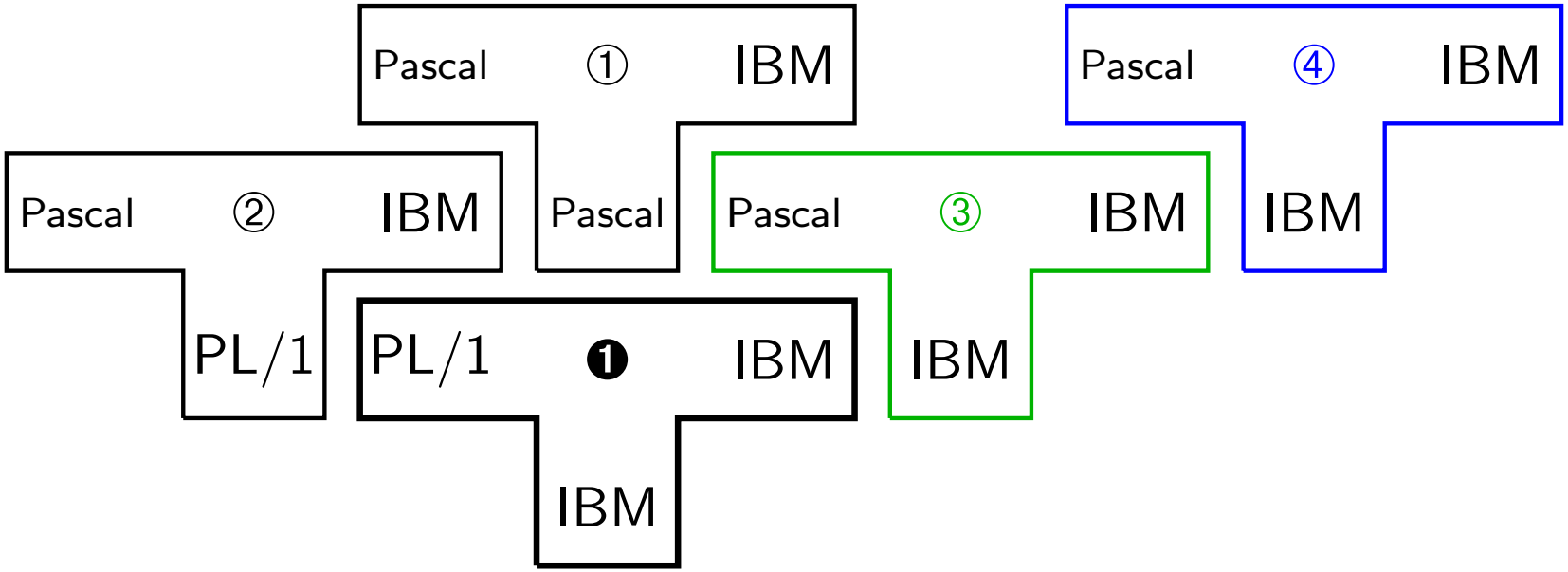
- Manuell:



und

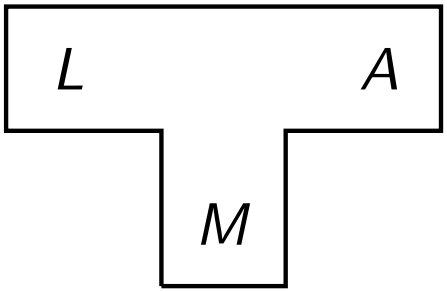


Portierung eines Compilers - Beispiel



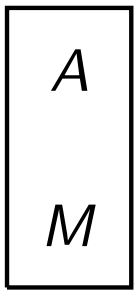
Portable Compiler

- Compiler übersetzt nicht direkt in M, sondern in Code für eine virtuelle Maschine A:

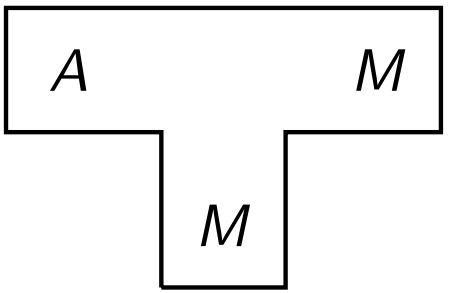


- Der Code wird ausgeführt durch einen

Interpretierer für A

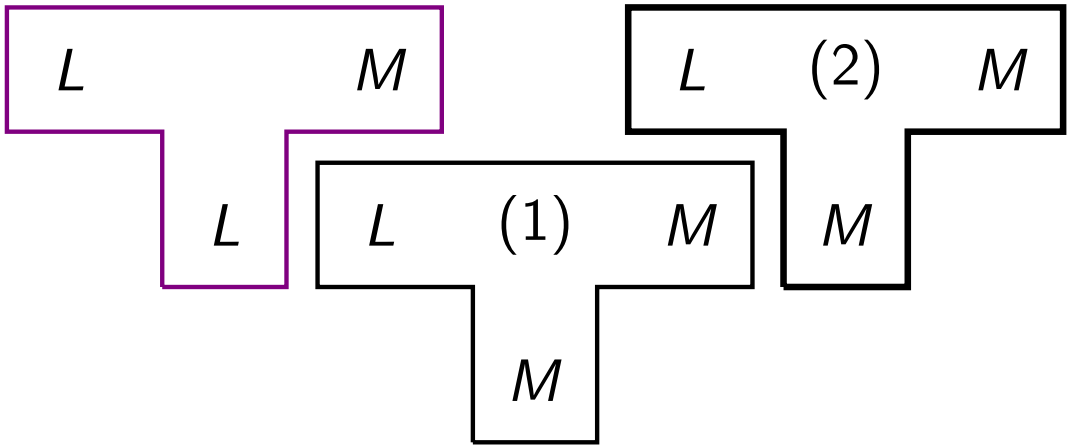


Compiler für A



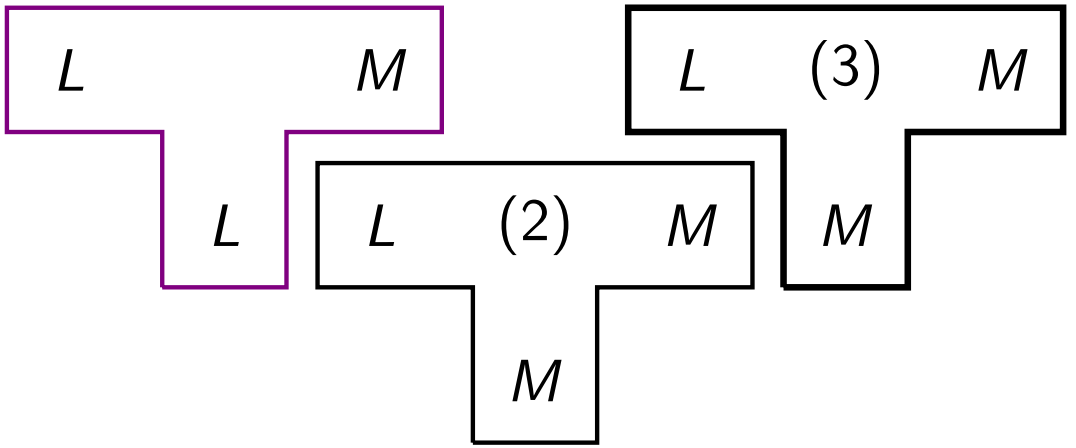
Testen eines Compilers (1)

- Ein guter Test für die Korrektheit eines neuen Compilers ist das folgende Vorgehen:



Testen eines Compilers (2)

- Nun wiederholt man den Vorgang mit dem neugewonnenen Compiler:



- Wenn nun die Compiler (2) und (3) identisch sind, haben wir ein gutes Indiz, dass unser Compiler korrekt arbeitet.