

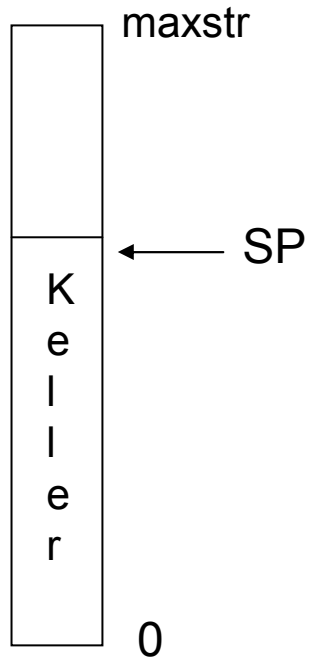
3. Sprachkonzepte und ihre Übersetzungen

- hier intuitiv **was** ein Compiler tut
- **Wie:** später
- Quellsprache: leicht abgemagertes Pascal
- Zielrechner: abstrakte Maschine, einfache Maschinensprache
- Sprach-Konzepte: Ausdrücke, Anweisungen, Kontrollstrukturen, Variablen, Felder, Verbunde, Zeiger, Prozeduren

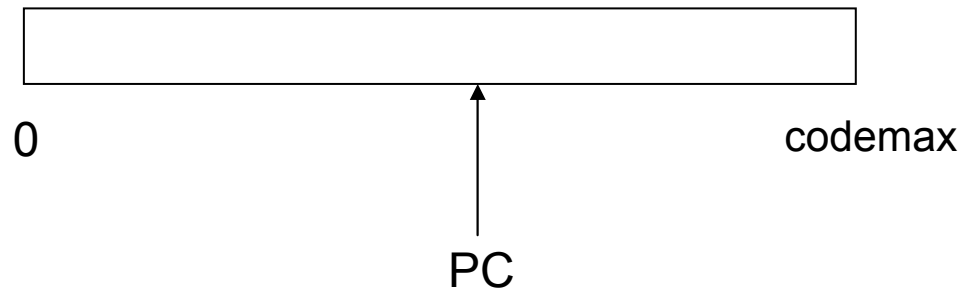
3. Sprachkonzepte und ihre Übersetzungen

Aufbau der P-Maschine:

Datenspeicher *STORE*



Programmspeicher *CODE*



3. Sprachkonzepte und ihre Übersetzungen

P-Befehle für **Ausdrücke** (Ausschnitt):

(N : numerischer Typ, T : bel. einfacher Typ, i : integer, r : real, b : boolean, a : Adresse)

Befehl	Bedeutung	Beding.	Ergebn.
add N	STORE[SP-1] := STORE[SP-1] + STORE[SP]; SP:= SP-1	(N,N)	(N)
sub N	STORE[SP-1] := STORE[SP-1] - STORE[SP]; SP:= SP-1	(N,N)	(N)
neg N	STORE[SP] := - STORE[SP]	(N)	(N)
and	STORE[SP-1] := STORE[SP-1] <i>and</i> STORE[SP]; SP:= SP-1	(b,b)	(b)
equ T	STORE[SP-1] := STORE[SP-1] = STORE[SP]; SP:= SP-1	(T,T)	(b)
geq T	STORE[SP-1] := STORE[SP-1] \geq STORE[SP]; SP:= SP-1	(T,T)	(b)

3. Sprachkonzepte und ihre Übersetzungen

P-Befehle zum Laden/Speichern:

Befehl	Bedeutung	Beding.	Ergebn.
ldo T q	SP:= SP+1; STORE[SP] := STORE[q]	$q \in$ [0,maxstr]	(T)
ldc T q	SP:= SP+1; STORE[SP] := q	Typ(q)= T	(T)
ind T	STORE[SP] := STORE[STORE[SP]]	(a)	(T)
sro T q	STORE[q] := STORE[SP]; SP:= SP-1;	(T) $q \in$ [0,maxstr]	
sto T	STORE[STORE[SP-1]] := STORE[SP]; SP:= SP-2;	(a, T)	

3. Sprachkonzepte und ihre Übersetzungen

P-Befehl bei **Wertzuweisung**:

Variablenbezeichnung

- links von Zuweisung : Adresse der ihr zugeordneten Zelle ?
(zum Überschreiben der alten Werte)
→ **L-Wert** der Variablen (*codeL*)
- auf rechter Seite: ihr Wert? (um Wert des Gesamtausdrucks zu berechnen)
→ **R-Wert** der Variablen (*codeR*)
- Übersetzung $x := y$, x, y Integer-Variablen:
 $codeL\ x\ p;$ (Wert auf Keller; verlängert Keller)
 $codeR\ y\ p;$ (Wert auf Keller; verlängert Keller)
 $sto\ i$
- p Funktion, $p(x)$ liefert Adresse von x