Übersetzung der allgemeinen Fallunterscheidung

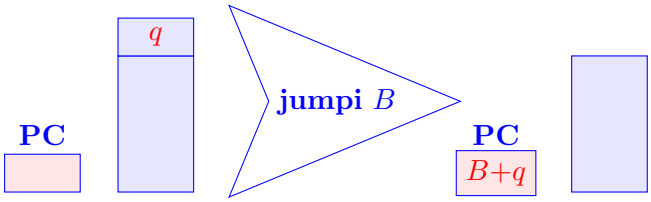
Eine **switch**-Anweisung ermöglicht eine **Vielfachverzweigung** abhängig vom Wert eines Selektorausdrucks e. Nehmen Sie zur Vereinfachung an, dass in der Anweisung nur zwischen den Fällen 0 bis $k-1$ für eine Konstante k ausgewählt wird. Für alle anderen Werte des Selektionsausdrucks gilt die **default**-Alternative. Die Fälle seien in aufsteigender Reihenfolge angeordnet und jeder Fall mit **break** abgeschlossen.

Die **switch**-Anweisung s hat das Aussehen:

switch e **of** {
case 0: ss_0 **break**;
case 1: ss_1 **break**;
⋮
case $k-1$: ss_{k-1} **break**;
default: ss_k
}

wobei die ss_i Folgen von Anweisungen sind.

switch-Anweisungen können mit indizierten Sprüngen übersetzt werden. Ein **indizierter Sprung** ist ein Sprung, bei dem die angegebene Zieladresse um einen unmittelbar vorher berechneten Wert erhöht wird. Wir benutzen dazu den Befehl **jumpi**.



$PC \leftarrow B + S[SP]; \quad SP{-}{-};$
Der indizierte Sprung **jumpi**.

Dann ergibt sich zunächst für die **switch**-Anweisung s die Befehlsfolge:

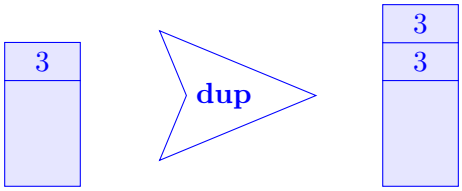
$\mathsf{code}^\rho\ s = \mathsf{code}^\rho_W\ e,$ **jumpi** B, B: **jump** C_0, **jump** C_1, …, **jump** C_k,
C_0: $\mathsf{code}^\rho\ ss_0$, **jump** D, C_1: $\mathsf{code}^\rho\ ss_1$, **jump** D, …, C_k: $\mathsf{code}^\rho\ ss_k$, D: …

Die Befehlsfolge führt einen indizierten Sprung in eine Tabelle durch, die ab Adresse B im Programmspeicher angelegt ist. Die **Sprungtabelle** enthält direkte Sprünge zu den jeweiligen Alternativen. Am Ende jeder Alternative steht ein Sprung hinter die **switch**-Anweisung.

Allerdings haben wir noch nicht überprüft, ob der Wert von e im Intervall $[0, k]$ liegt; das wird vom Makro **check** k B überprüft. Das Makro **check** ist eine Abkürzung für:

check k B ≡ **dup**, **loadc** 0, **geq**, **jumpz** A, **dup**, **loadc** k, **leq**, **jumpz** A, **jumpi** B,
A: **alloc** -1, **loadc** k, **jumpi** B

Wir müssen den ganzzahligen Wert oben auf dem Keller, den wir zur Fallunterscheidung benutzen, überprüfen, ob er innerhalb des Intervalls $[0, k]$ liegt. Nur dann können wir ihn zur Indizierung der Sprungtabelle benutzen.



$S[SP+1] \leftarrow S[SP]; \quad SP{+}{+};$
Der Befehl **dup**.

Dieser Wert wird in zwei Vergleichen benötigt. Da unsere Maschine den Wert bei jedem Vergleich konsumiert, **duplizieren** wir ihn vorher. Dazu dient der Befehl **dup**.

Die Idee zur Implementierung des Makro ist dann einfach: Für Werte e im Intervall $[0, k-1]$ führt der indizierte Sprung zum unbedingten Sprung an die Anfangsadresse der e-ten Alternative. Ist der Wert $e < 0$ oder $e > k$, ersetzen wir ihn vor dem indizierten Sprung durch k, so dass der indizierte Sprung zum unbedingten Sprung auf die **default**-Alternative führt.

Die **switch**-Anweisung s erzeugt also die Befehlsfolge:

$\mathsf{code}^\rho\ s = \mathsf{code}^\rho_W\ e,\quad$ **check** k B, $\quad$ B: **jump** C_0, **jump** C_1, …, **jump** C_k,
C_0: $\mathsf{code}^\rho\ ss_0$, **jump** D, $\quad C_1$: $\mathsf{code}^\rho\ ss_1$, **jump** D, $\quad \ldots, \quad C_k$: $\mathsf{code}^\rho\ ss_k$, $\quad$ D: …

Die stark vereinfachte **switch**-Anweisung lässt sich verallgemeinern:

- Ist der kleinste vorkommende Selektorwert u (statt 0), vermindern wir den Wert von e um u, bevor wir ihn als Index benutzen.
- Eine aufsteigende Anordnung der Selektorwerte ist nicht erforderlich.
- Auch Lücken im Intervall der möglichen Selektorwerte dürfen vorkommen. Fehlende Einträge der Sprungtabelle werden dann mit unbedingten Sprüngen auf die **default**-Alternative aufgefüllt. Probleme treten auf, wenn das Intervall der möglichen Selektorwerte sehr groß ist, aber nur sehr wenige Werte vorkommen. Dann kann eine geschachtelte **if-then-else**-Anweisung effizienter sein.
- Man erzeugt die Programmfragmente $\mathsf{code}^\rho\ ss_i$ in der Reihenfolge des Programmtextes. Fehlt dann ein **break**, kann man den zugehörigen Sprung **jump** D einfach weglassen.

Beispiel:

$s \equiv$ **switch** e **of** {
case 5: ss_5 ~~**break**;~~
case 2: ss_2 **break**;
case 4: ss_4 **break**;
default: ss_6
}

Kleinster Selektorwert ist 2, größter 5; default-Wert wird 6: Differenz zum kleinsten Wert 4.

$\mathsf{code}^\rho\ s = \mathsf{code}^\rho_W\ e,\quad \overbrace{\textbf{loadc}\ 2,\ \textbf{sub},}^{\text{kleinsten Selektorwert abziehen}}\quad$ **check** 4 B,
B: **jump** C_2, **jump** C_6, **jump** C_4, **jump** C_5, **jump** C_6,
C_5: $\mathsf{code}^\rho\ ss_5$, ~~**jump** D,~~ C_2: $\mathsf{code}^\rho\ ss_2$, **jump** D, C_4: $\mathsf{code}^\rho\ ss_4$, **jump** D,
C_6: $\mathsf{code}^\rho\ ss_6$,
D: …

Die Sprungtabelle enthält der Reihe nach die Sprungziele für Selektorwerte 2, 3, 4, 5, default.