

Visibly Pushdown Automata

Proseminar WS 16/17

Albert-Ludwigs-Universität Freiburg

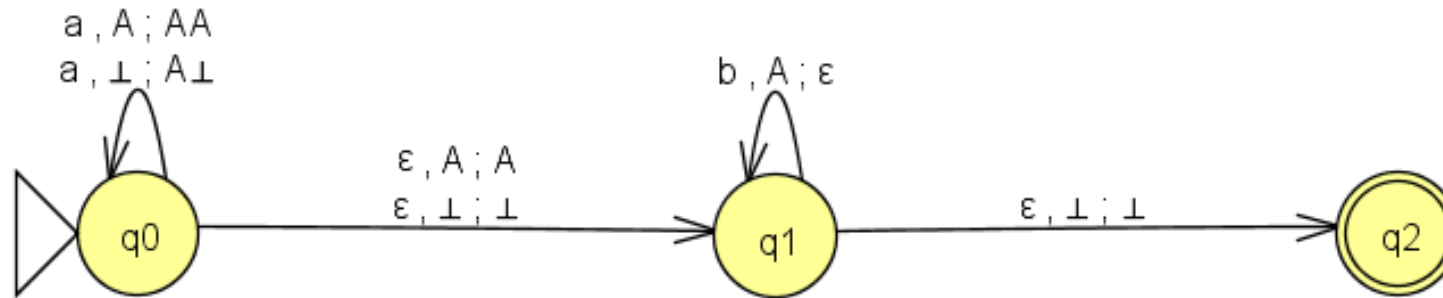
Gliederung

- Kellerautomaten
- Visibly Pushdown Automata (VPA)
- Visibly Pushdown Language (VPL)
- Abschlusseigenschaften

Kellerautomaten

- $M = (Q, \Sigma, \Gamma, \delta, q_0, \perp, Q_F)$ mit:
 - Q – eine endliche Menge an Zuständen
 - Σ – ein endliches Eingabealphabet
 - Γ – endliches Kelleralphabet
 - δ – eine Zustandsübergangsfunktion
 - $q_0 \in Q$ – der Startzustand
 - $\perp \in \Gamma$ – Anfangssymbol im Keller
 - $Q_F \subseteq Q$ – eine endliche Menge an Endzuständen

Kellerautomaten - Beispiel



$$L = \{a^n b^n \mid n \in \mathbb{N}_0\}$$

kontextfreie Sprachen (CFL)

- Erkennen von Klammersausdrücken

Visibly Pushdown Automata (VPA) - Definition

- $M = (Q, \tilde{\Sigma}, \Gamma, \delta, q_0, \perp, Q_F)$ mit $\tilde{\Sigma} = (\Sigma_{call}, \Sigma_{ret}, \Sigma_{int})$:
 - Q – eine endliche Menge an Zuständen
 - Γ – Kelleralphabet
 - $\perp \in \Gamma$ – Anfangssymbol im Keller
 - $q_0 \in Q$ – eine endliche Menge an Startzuständen
 - δ – eine Zustandsübergangsfunktion
 - $Q_F \subseteq Q$ – eine endliche Menge an möglichen Endzuständen
- Kellerautomat: $M = (Q, \Sigma, \Gamma, \delta, q_0, \perp, Q_F)$

VPA - Eingabealphabet

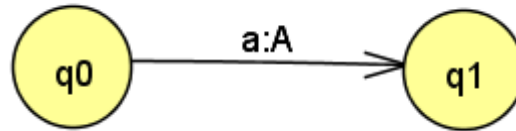
- $\tilde{\Sigma} = (\Sigma_{call}, \Sigma_{ret}, \Sigma_{int})$
- Σ_{call} – ein endliches Alphabet für *Call Aktionen*
- Σ_{ret} – ein endliches Alphabet für *Return Aktionen*
- Σ_{int} ein endliches Alphabet für *interne Aktionen*

- **Push** – Schreibt Zeichen in Keller, Zustandsübergang
- **Pop** – Liest (Löscht) Zeichen aus dem Keller, Zustandsübergang
- **Internal** – Zustandsübergang

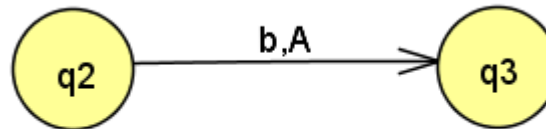
VPA - Transitionen

- Für $\Sigma_{call} = \{a\}$, $\Sigma_{ret} = \{b\}$, $\Sigma_{int} = \{c\}$

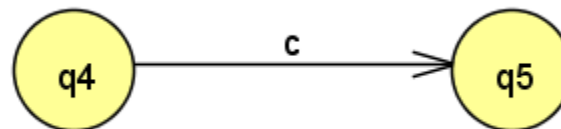
- Call-Transition



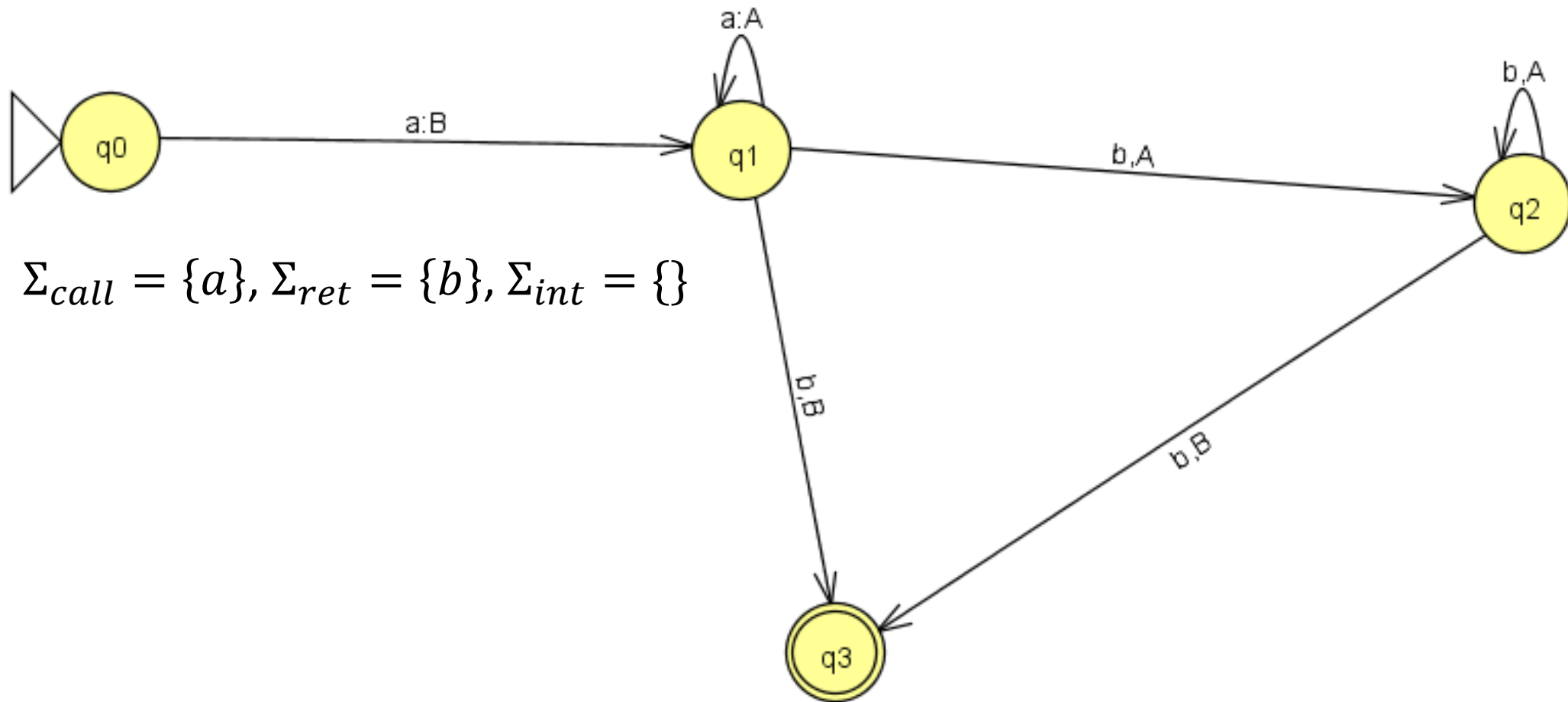
- Pop-Transition



- Internal-Transition

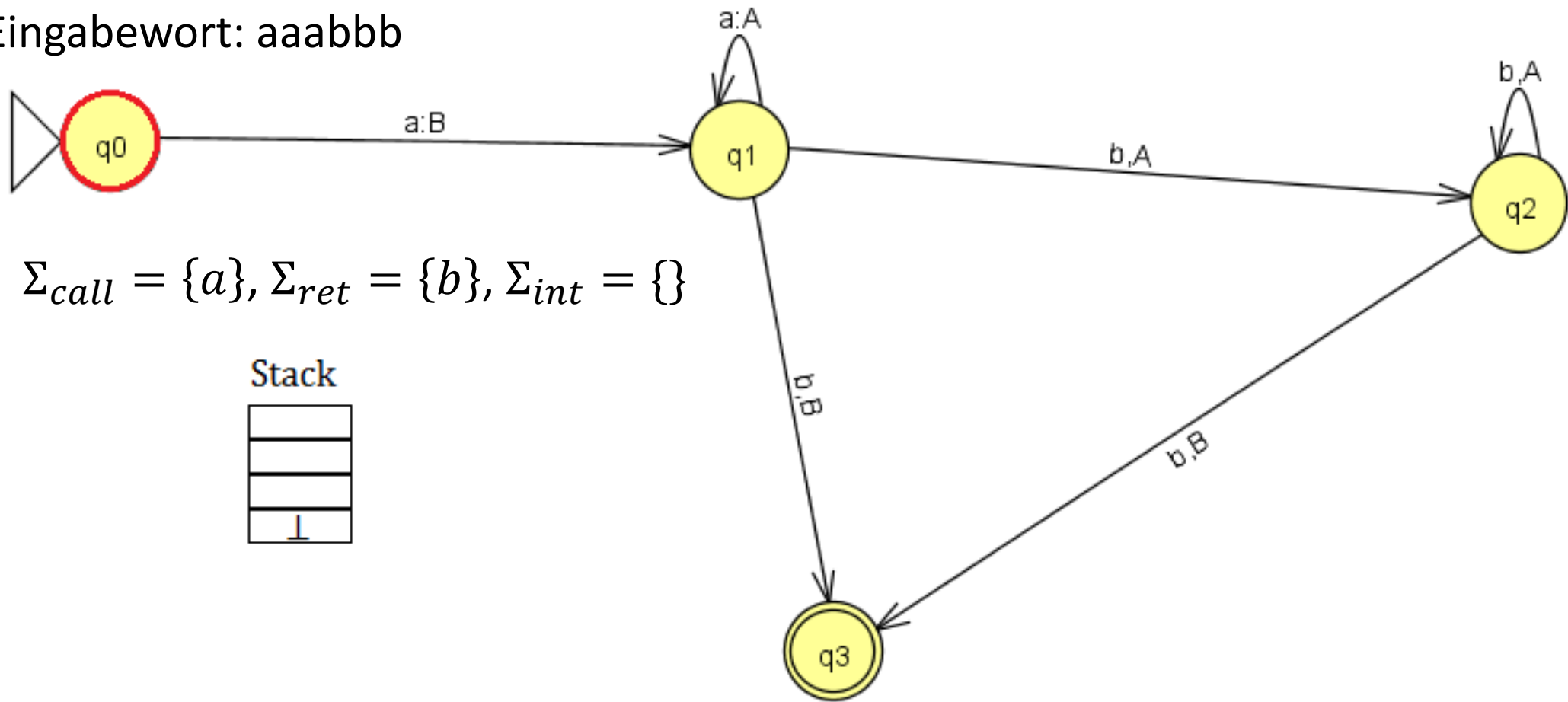


VPA - Beispiel



VPA - Beispiel

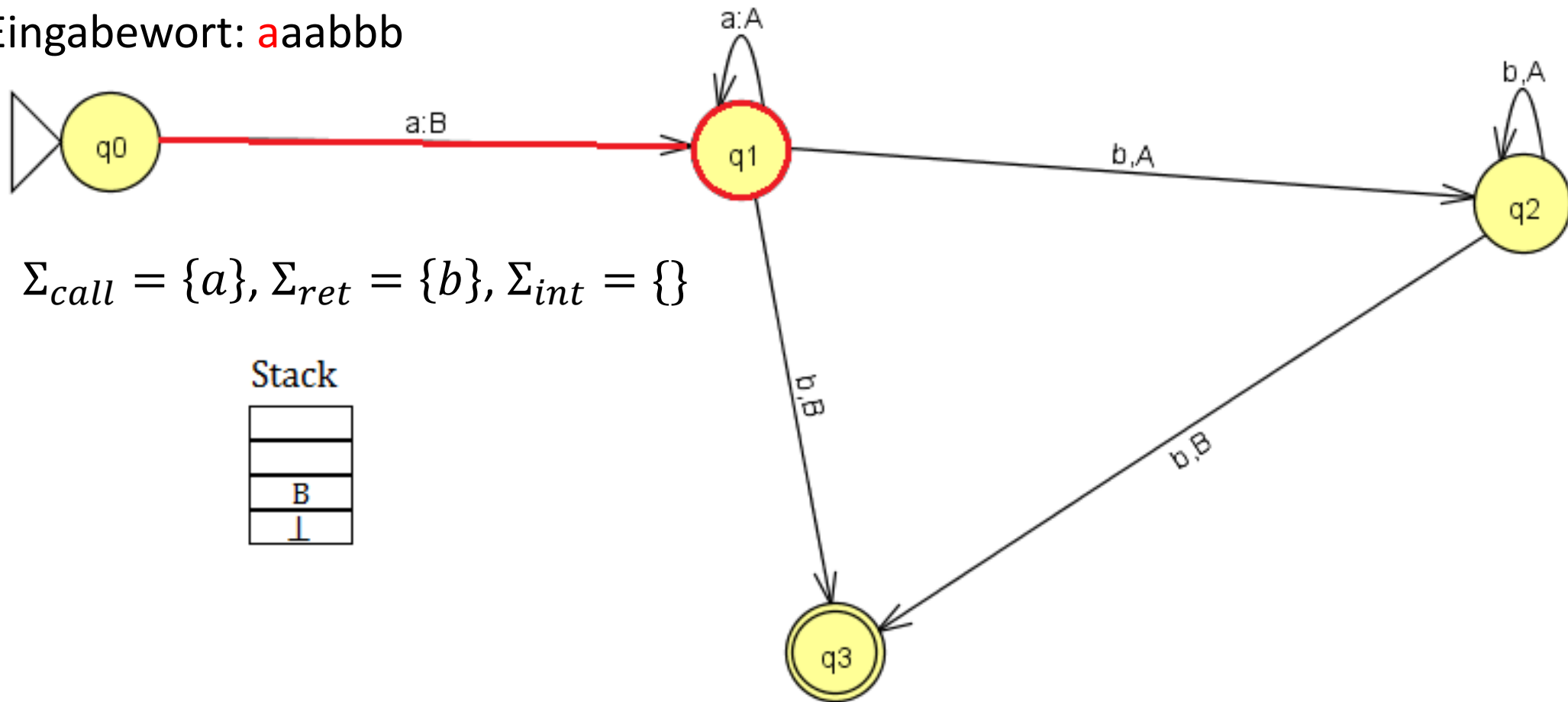
Eingabewort: aaabbb



$$\Sigma_{call} = \{a\}, \Sigma_{ret} = \{b\}, \Sigma_{int} = \{\}$$

VPA - Beispiel

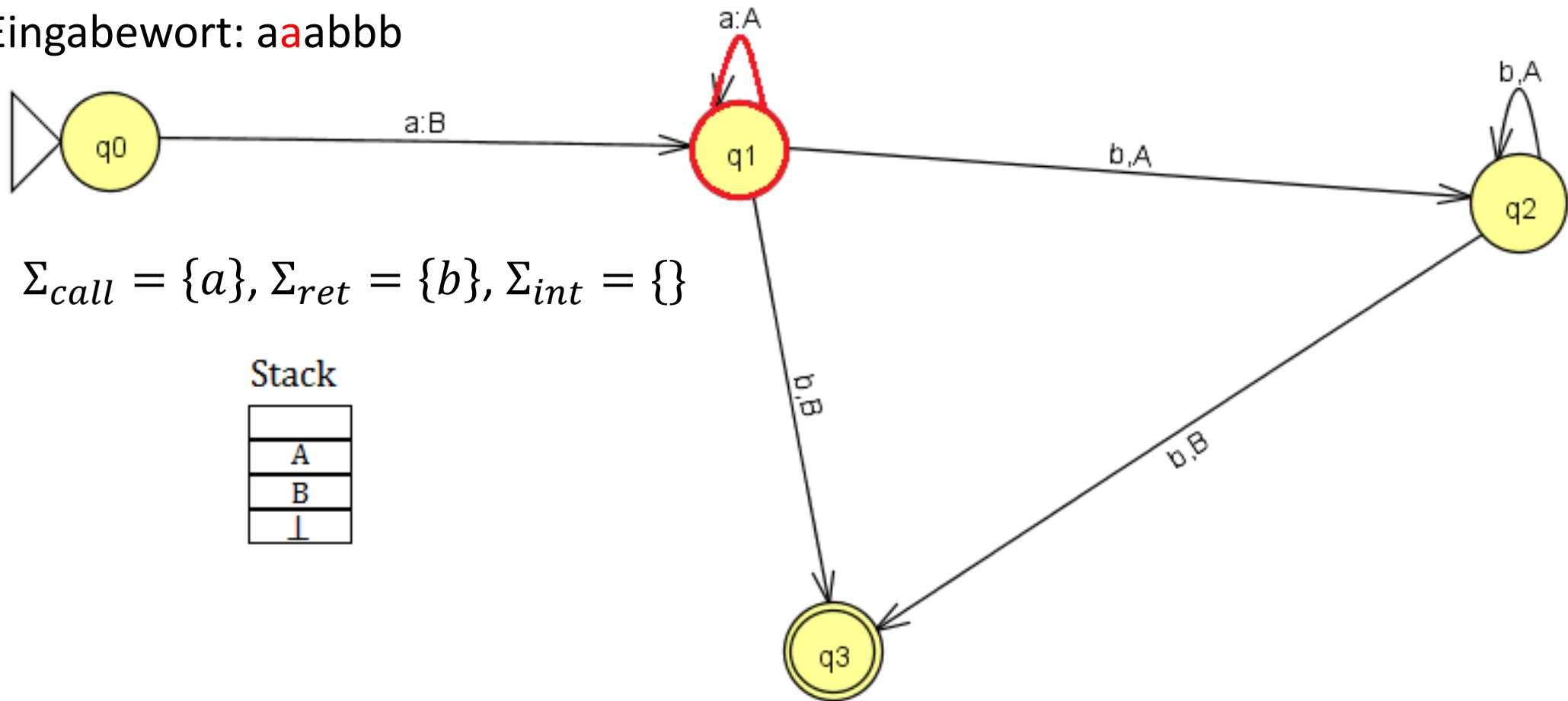
Eingabewort: **a**aabbb



$$\Sigma_{call} = \{a\}, \Sigma_{ret} = \{b\}, \Sigma_{int} = \{\}$$

VPA - Beispiel

Eingabewort: a**a**bbbb



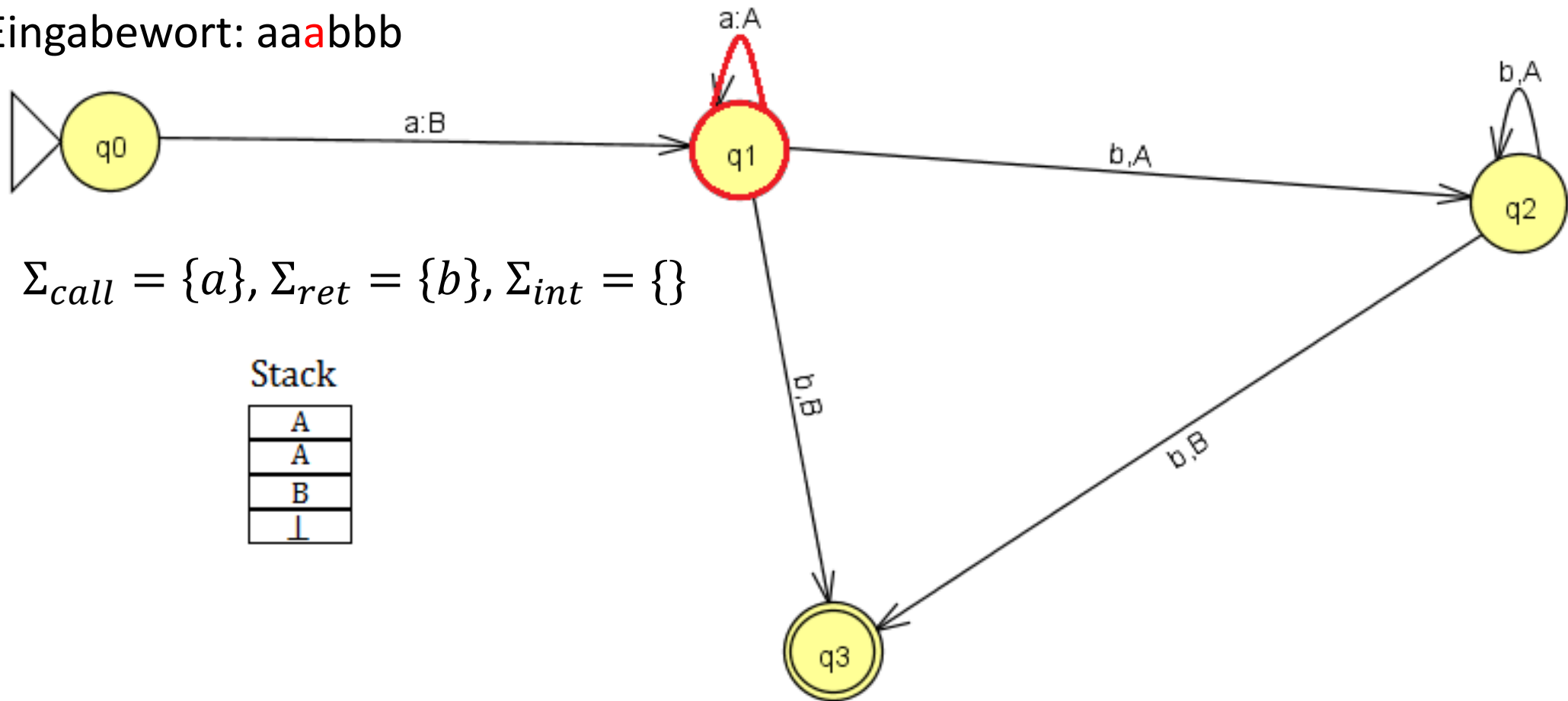
$$\Sigma_{call} = \{a\}, \Sigma_{ret} = \{b\}, \Sigma_{int} = \{\}$$

Stack

A
B
I

VPA - Beispiel

Eingabewort: aa**a**bbb



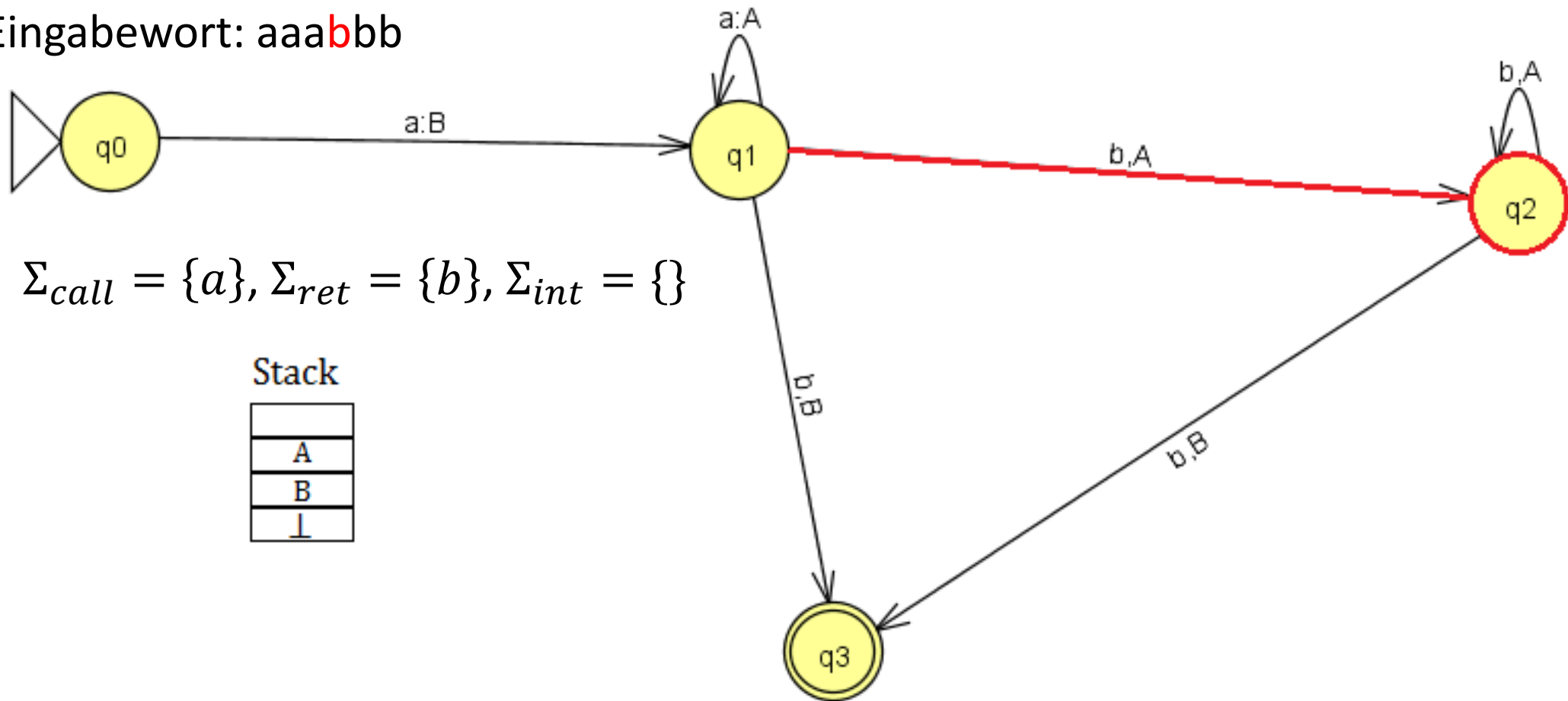
$$\Sigma_{call} = \{a\}, \Sigma_{ret} = \{b\}, \Sigma_{int} = \{\}$$

Stack

A
A
B
I

VPA - Beispiel

Eingabewort: aaa**b**bb



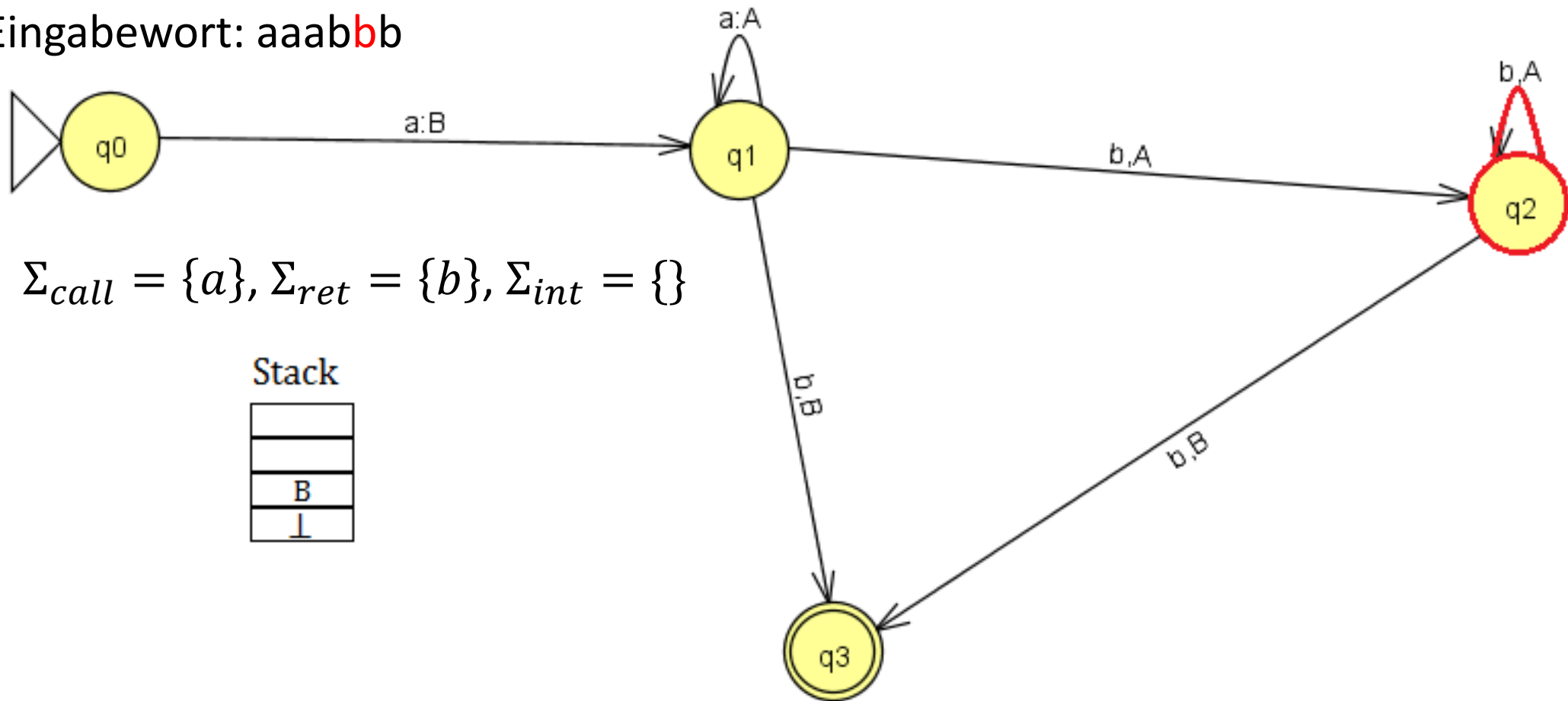
$$\Sigma_{call} = \{a\}, \Sigma_{ret} = \{b\}, \Sigma_{int} = \{\}$$

Stack

A
B
$\perp$

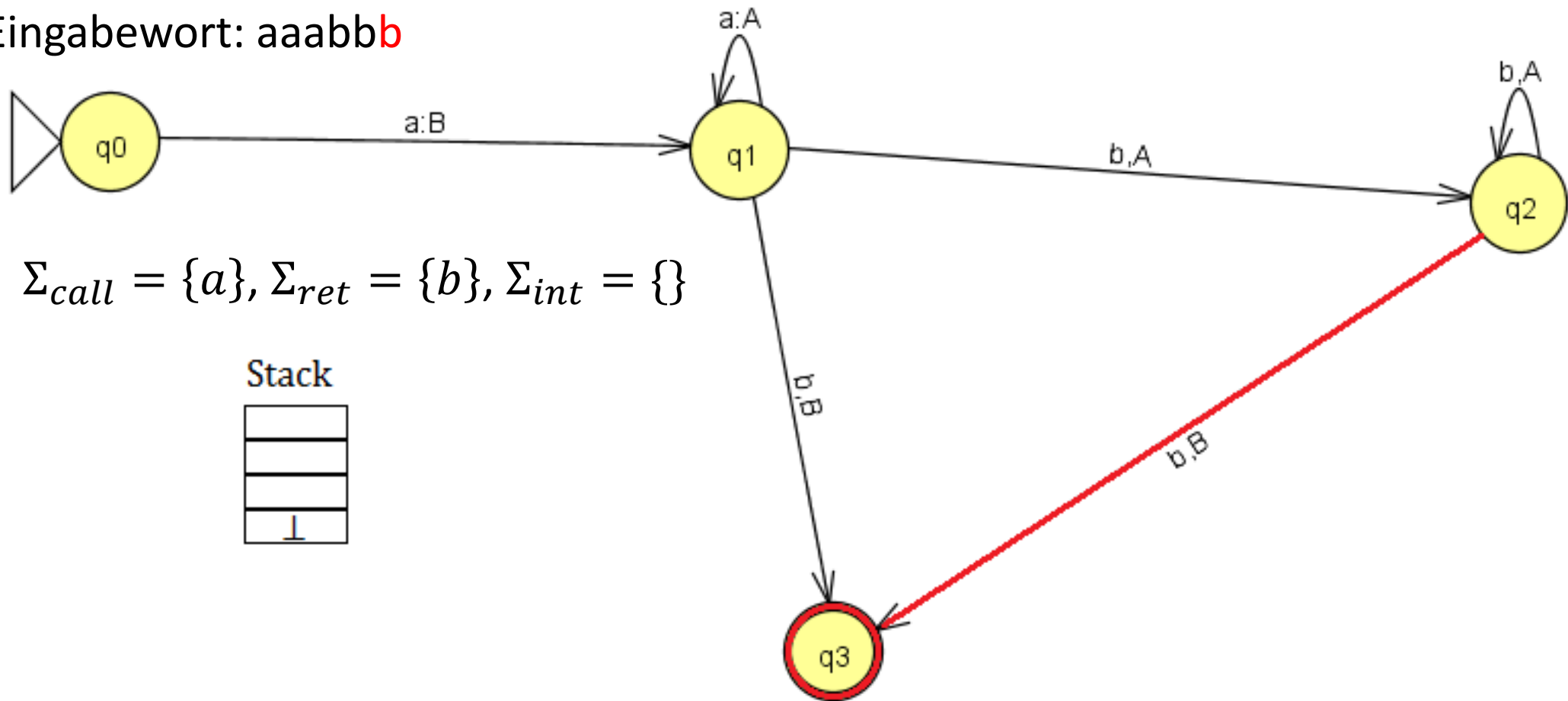
VPA - Beispiel

Eingabewort: aaab**b**b



VPA - Beispiel

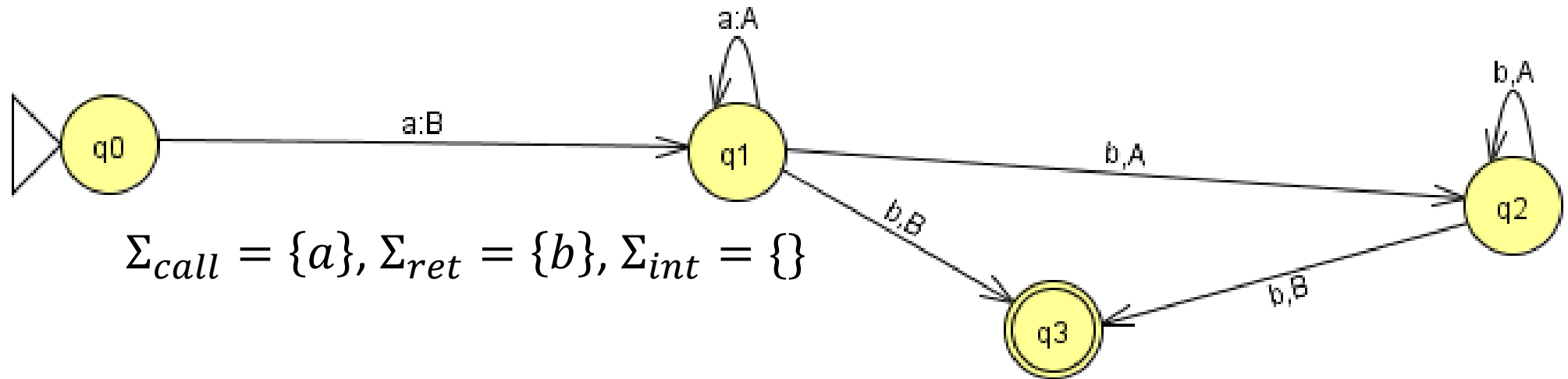
Eingabewort: aaabb**b**



Visibly Pushdown Languages (VPL)

- Eine *Visibly Pushdown Language (VPL)* ist eine Sprache, für die ein VPA M existiert, dass gilt: $L(M) = L$

- M :



- M erkennt $L = \{a^n b^n | n \in \mathbb{N} \setminus 0\}$

VPL - Eigenschaften

- Mächtiger als reguläre Sprachen
 - Regulär $\subset$ VPL $\subset$ CFL
- $L = \{a^n b a^n \mid n \in \mathbb{N}\}$ nicht möglich
- Determinisierung möglich
- Eigenschaften von regulären Sprachen

VPL - Abschlusseigenschaften im Vergleich

	Closure under				
	$\cup$	$\cap$	Complement	Concat.	Kleene-*
Regular	Yes	Yes	Yes	Yes	Yes
CFL	Yes	No	No	Yes	Yes
DCFL	No	No	Yes	No	No
VPL	Yes	Yes	Yes	Yes	Yes

VPL -Entscheidungsprobleme im Vergleich

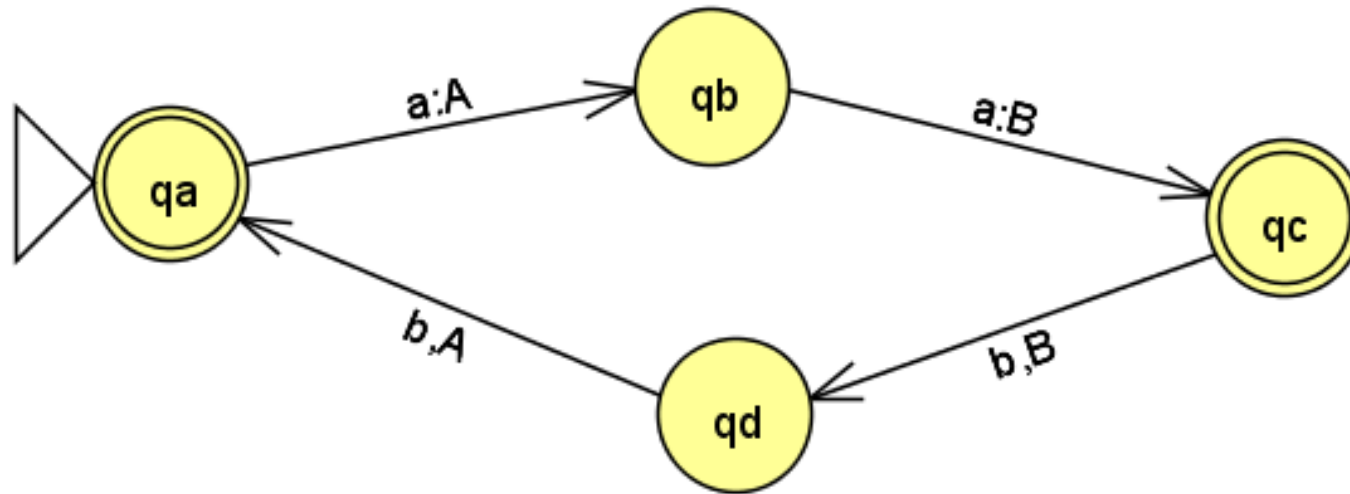
	Decision problems for automata		
	Emptiness	Univ./Equiv.	Inclusion
Regular	NLOG	PSPACE	PSPACE
CFL	P TIME	Undecidable	Undecidable
DCFL	P TIME	Decidable	Undecidable
VPL	P TIME	EXPTIME	EXPTIME

=> Gleiche Eigenschaften wie Reguläre Sprachen, aber ausdrucksmächtiger

VPL - Abschluss unter Schnitt

$$\Sigma_{call} = \{a\}, \Sigma_{ret} = \{b\}$$

M_1 :

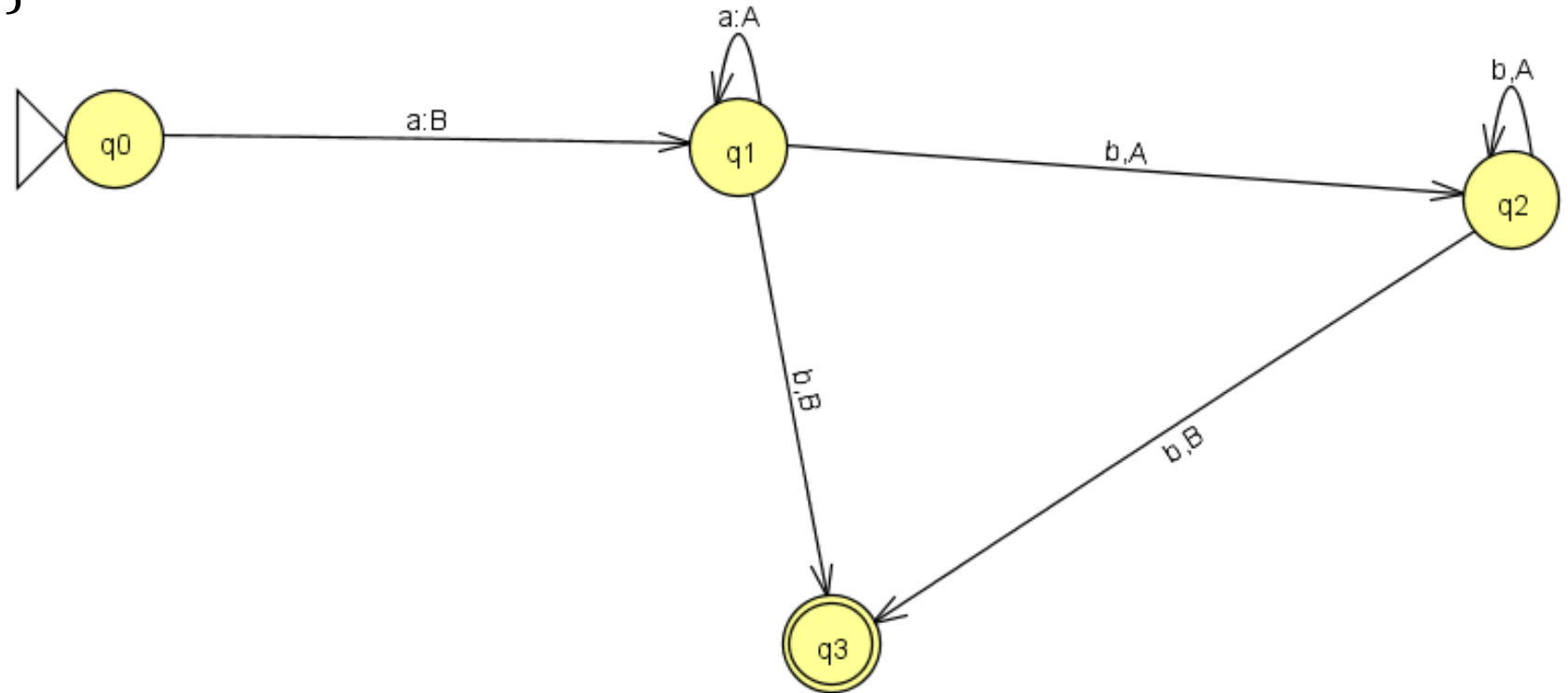


$$L_1 = \{(aabb)^*(aa)^m \mid m \in \{0,1\}, n \in \mathbb{N}_0\}$$

VPL - Abschluss unter Schnitt

$$\Sigma_{call} = \{a\}, \Sigma_{ret} = \{b\}$$

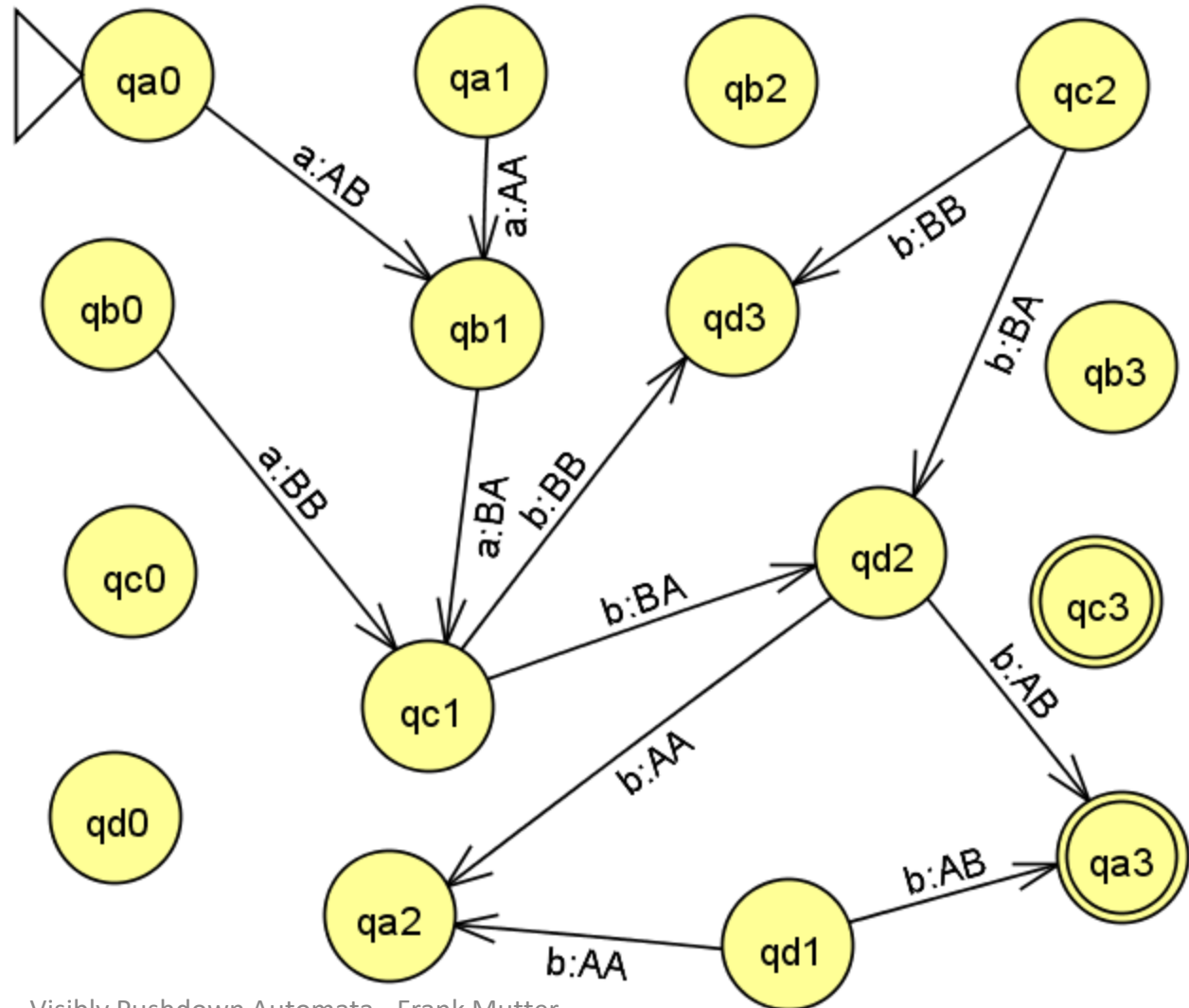
M_2 :



$$L_2 = \{a^n b^n \mid n \in \mathbb{N} \setminus 0\}$$

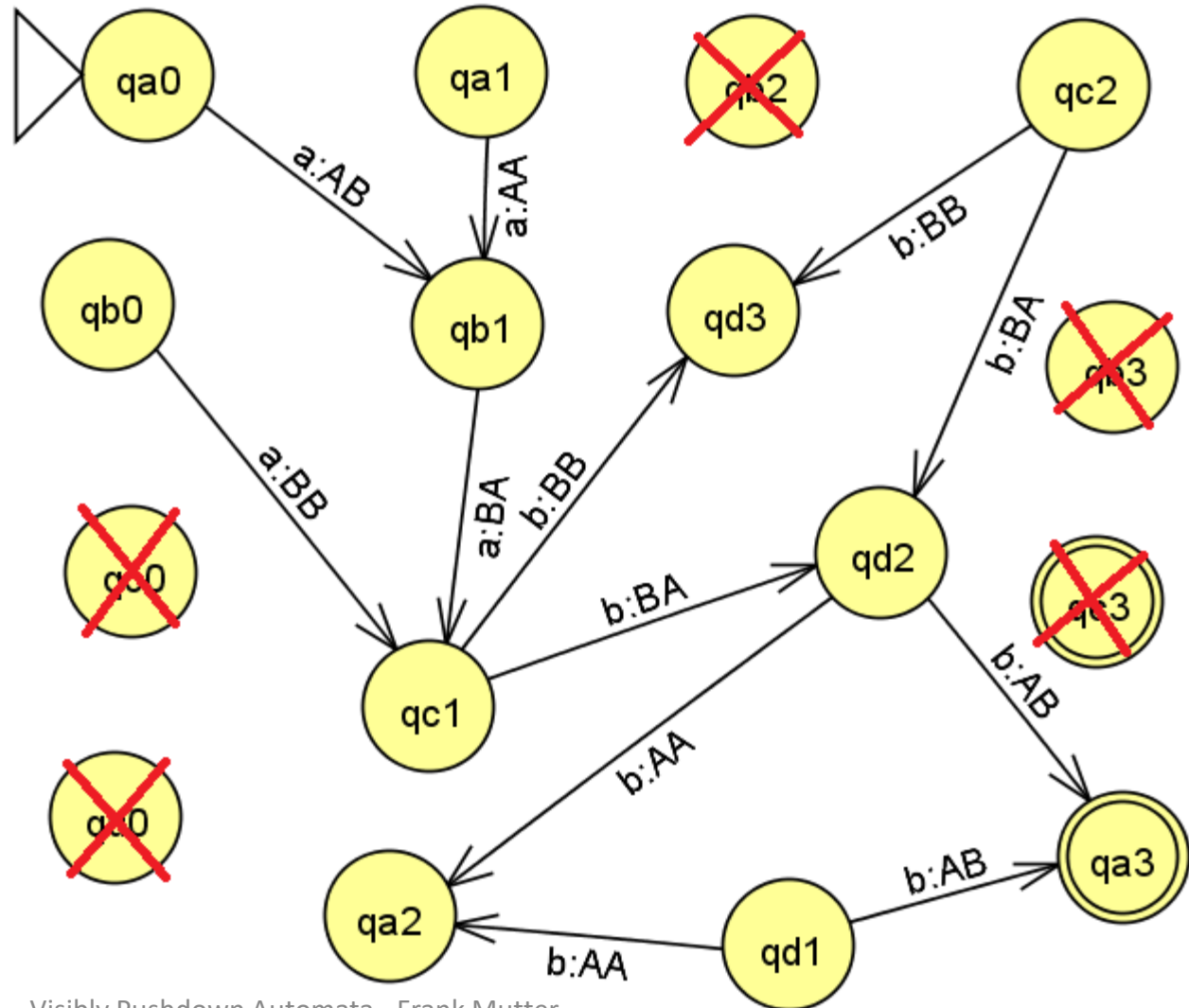
VPL - Abschluss unter Schnitt

$$\Sigma_{call} = \{a\}, \Sigma_{ret} = \{b\}$$



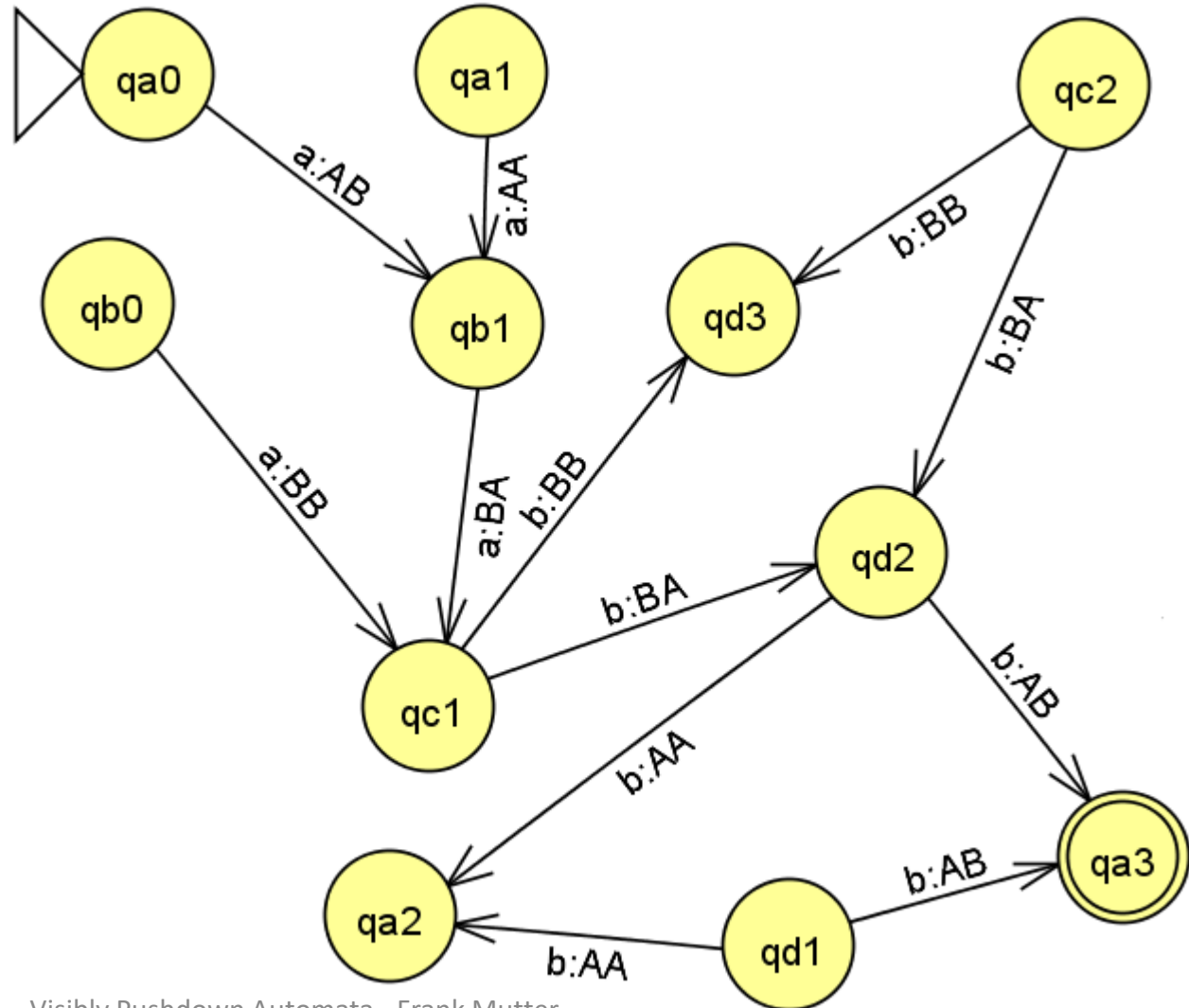
VPL - Abschluss unter Schnitt

$$\Sigma_{call} = \{a\}, \Sigma_{ret} = \{b\}$$



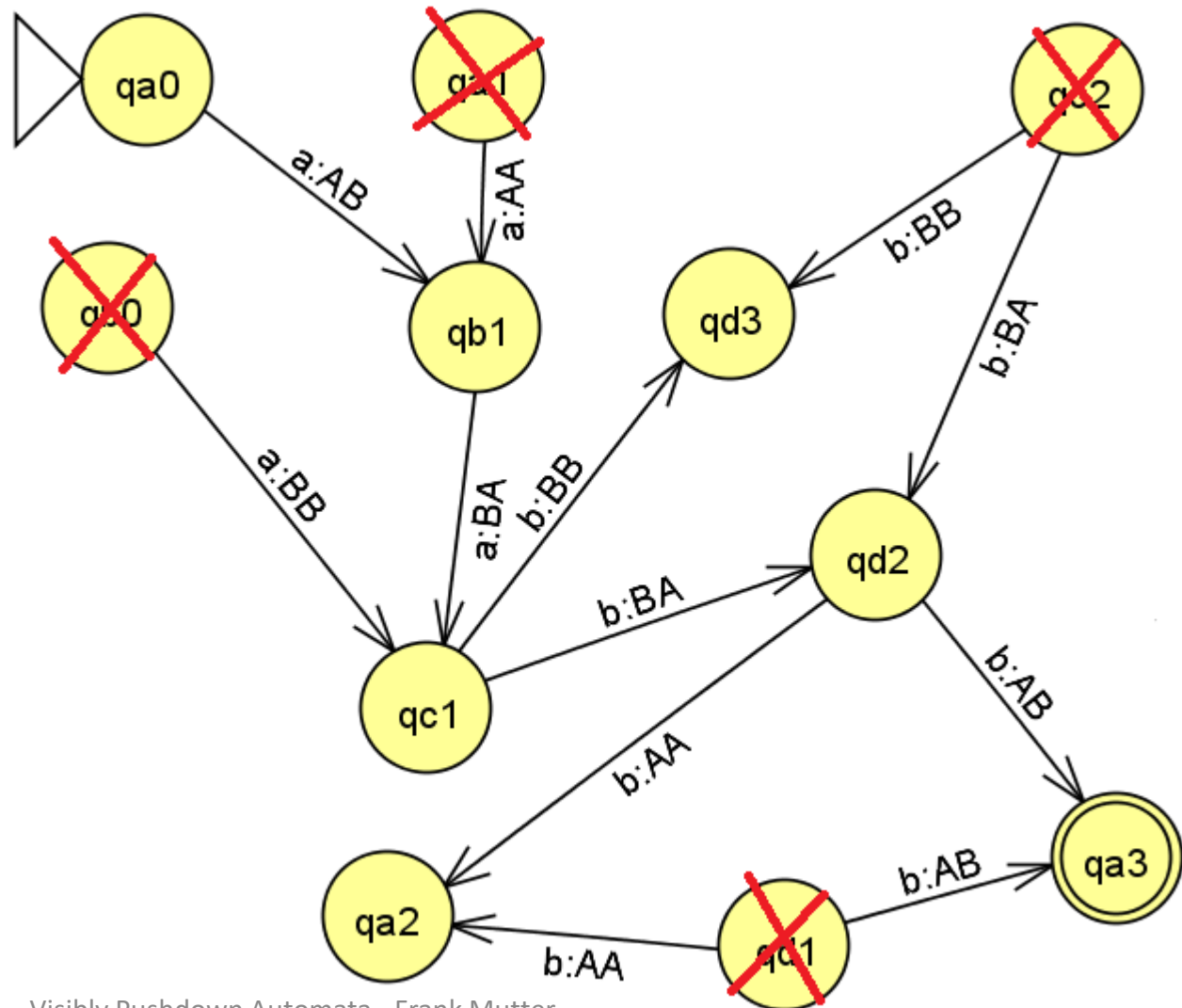
VPL - Abschluss unter Schnitt

$$\Sigma_{call} = \{a\}, \Sigma_{ret} = \{b\}$$



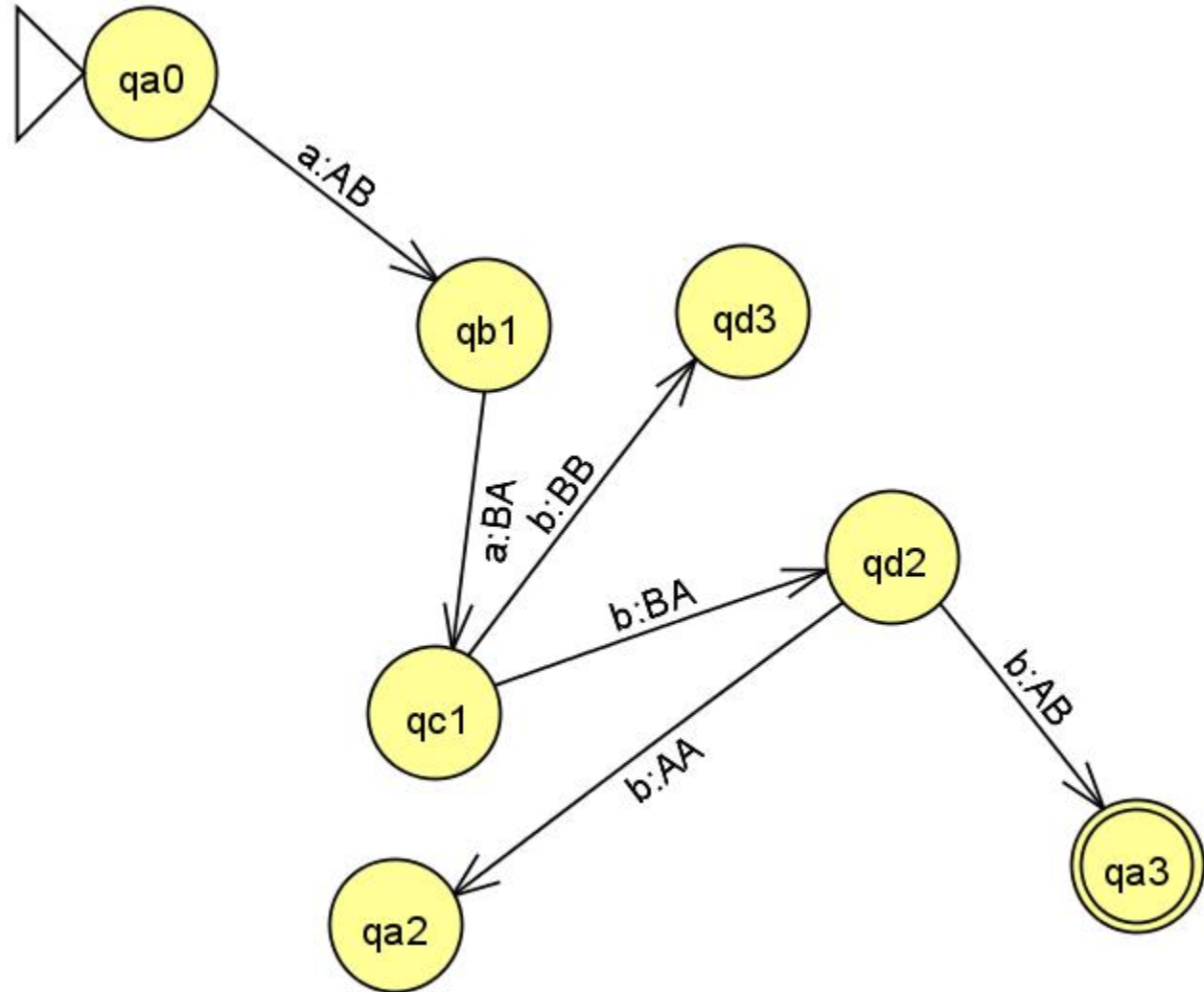
VPL - Abschluss unter Schnitt

$$\Sigma_{call} = \{a\}, \Sigma_{ret} = \{b\}$$



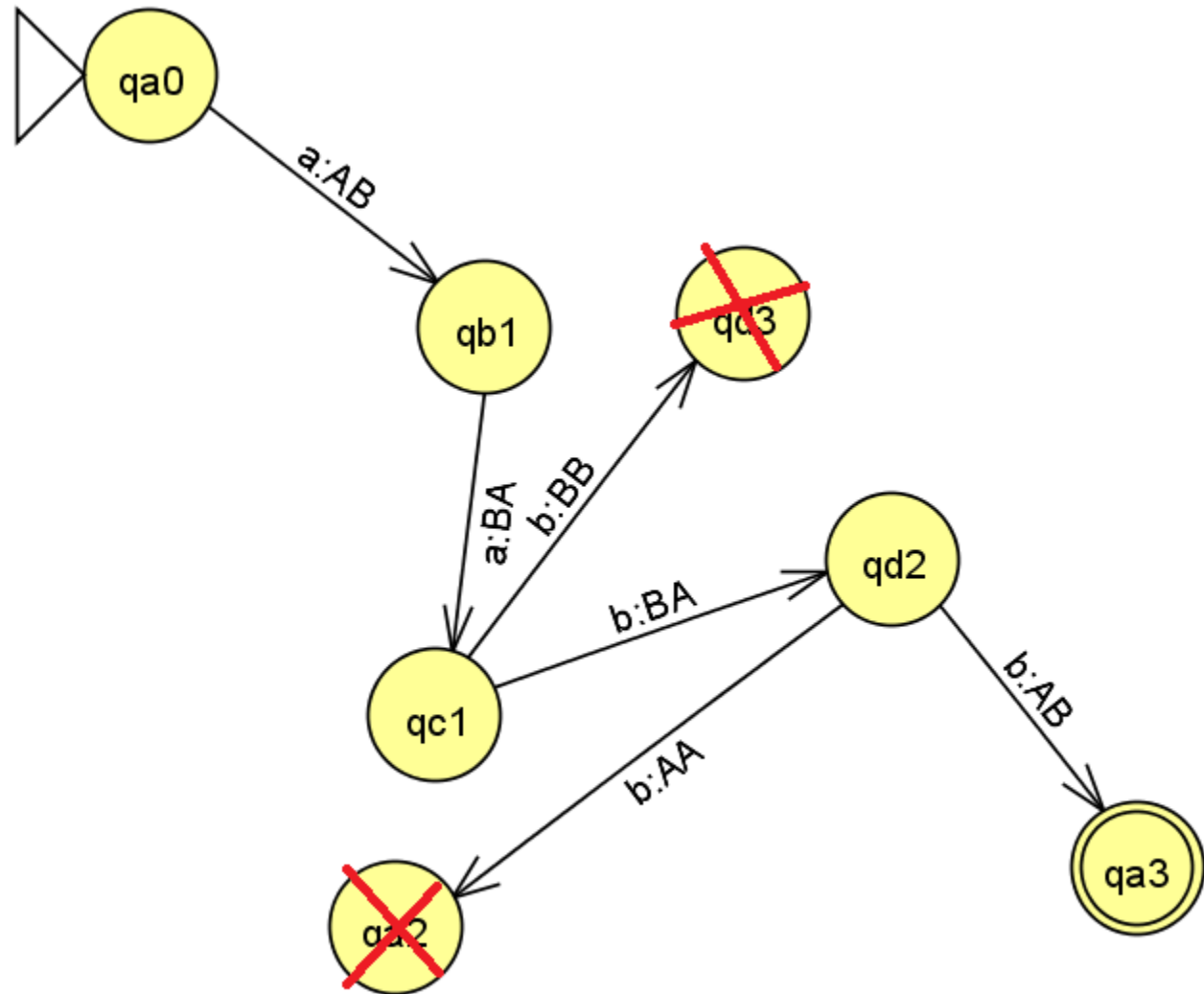
VPL - Abschluss unter Schnitt

$$\Sigma_{call} = \{a\}, \Sigma_{ret} = \{b\}$$



VPL - Abschluss unter Schnitt

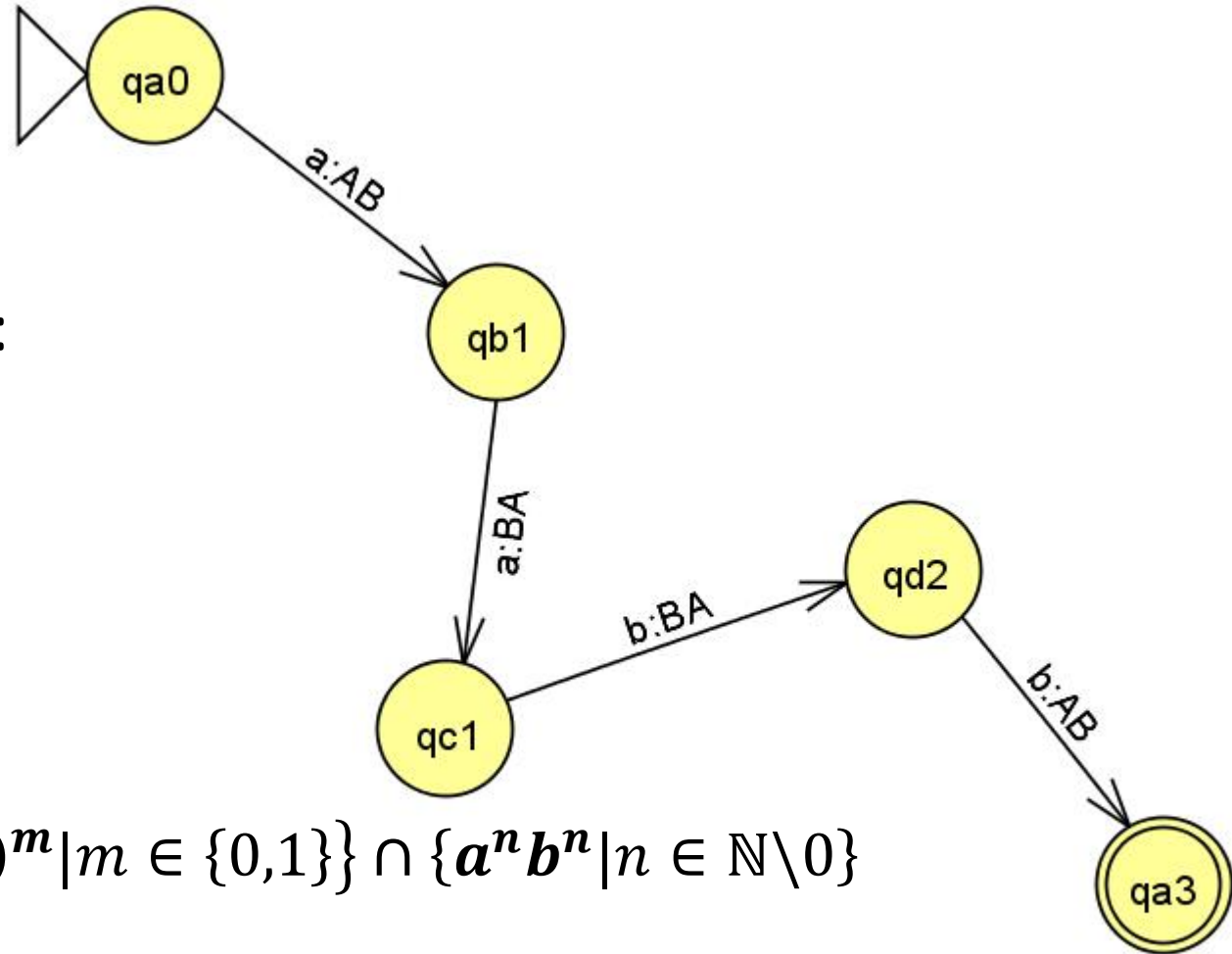
$$\Sigma_{call} = \{a\}, \Sigma_{ret} = \{b\}$$



VPL - Abschluss unter Schnitt

$$\Sigma_{call} = \{a\}, \Sigma_{ret} = \{b\}$$

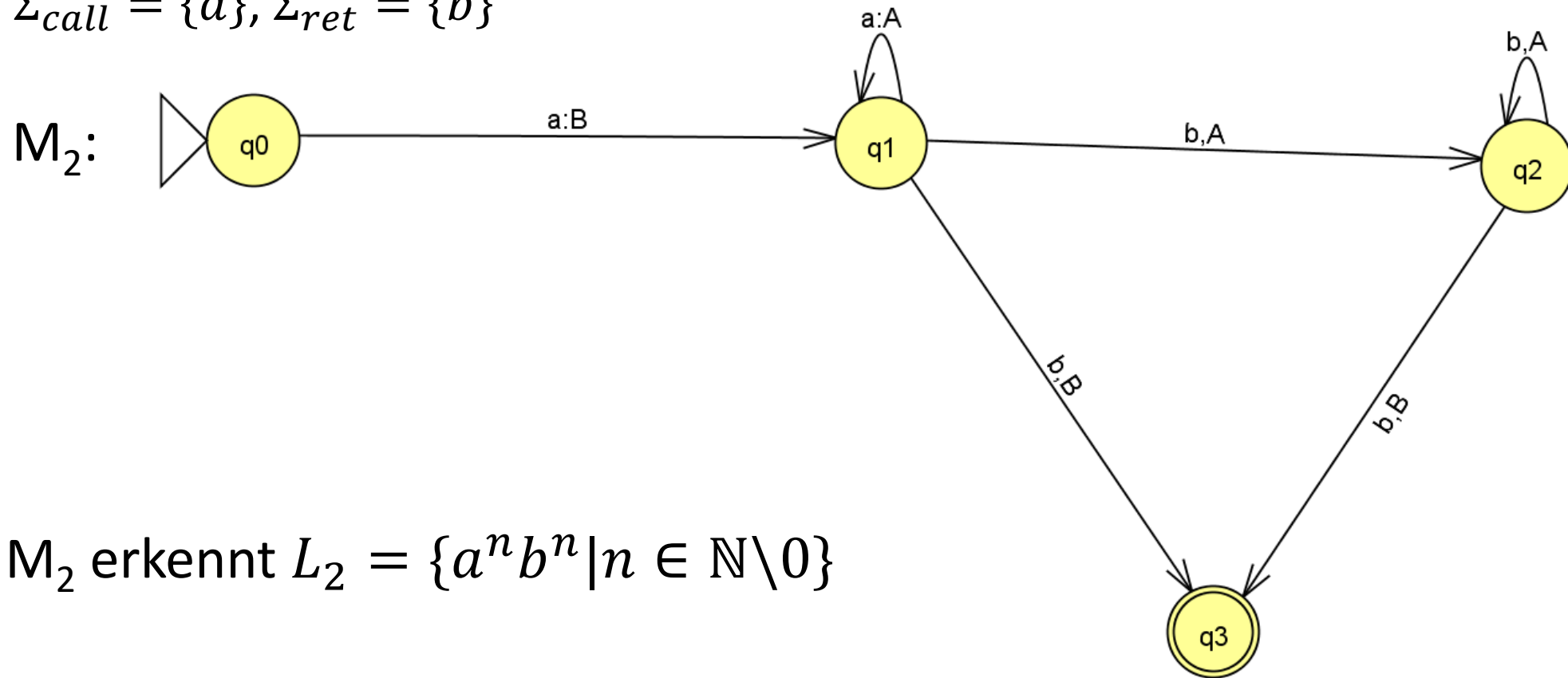
M_3 :



$$L_3 = L_1 \cap L_2 = \{(aabb)^*(aa)^m \mid m \in \{0,1\}\} \cap \{a^n b^n \mid n \in \mathbb{N} \setminus \{0\}\} \\ = \{aabb\}$$

VPL - Komplement

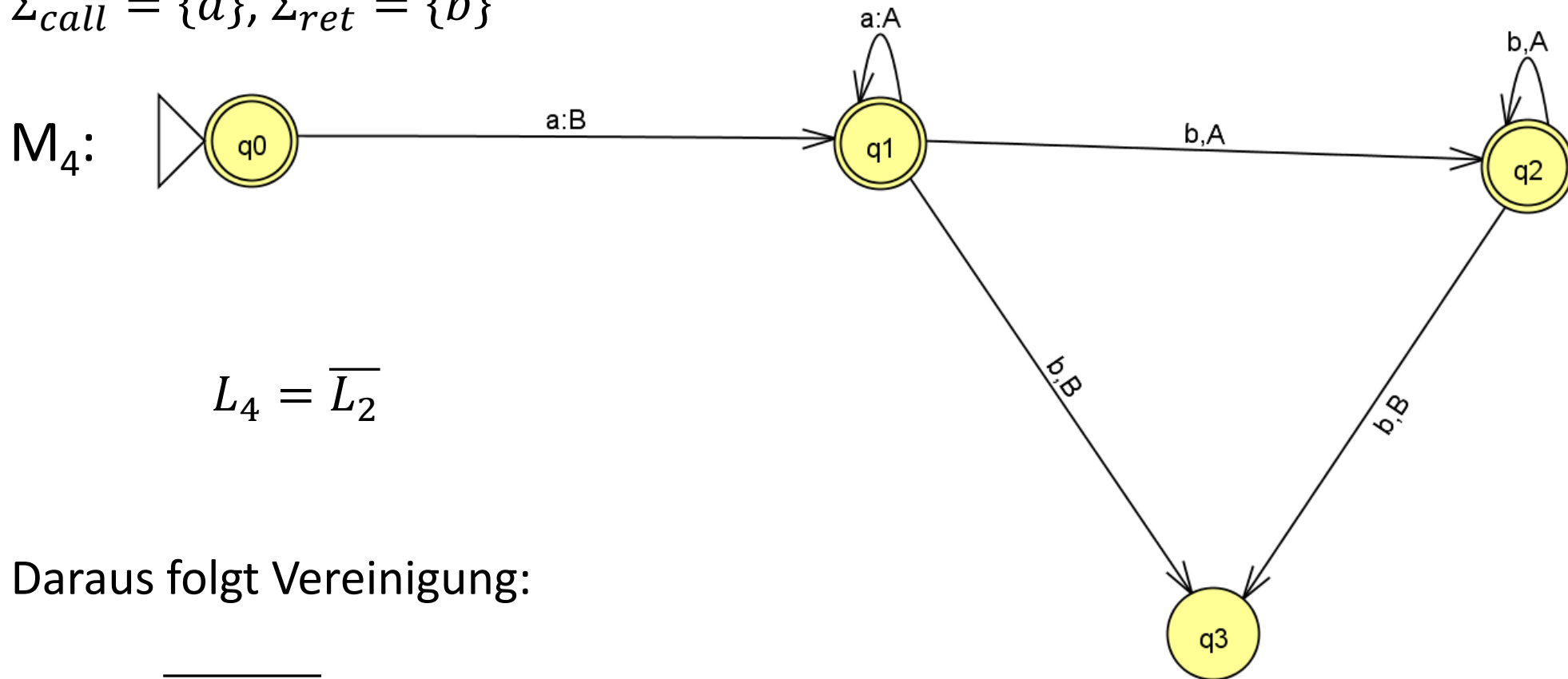
$$\Sigma_{call} = \{a\}, \Sigma_{ret} = \{b\}$$



- M_2 erkennt $L_2 = \{a^n b^n | n \in \mathbb{N} \setminus \{0\}\}$

VPL - Komplement

$$\Sigma_{call} = \{a\}, \Sigma_{ret} = \{b\}$$



$$L_4 = \overline{L_2}$$

Daraus folgt Vereinigung:

$$L_5 = \overline{\overline{L_1} \cap \overline{L_2}} = L_1 \cup L_2$$

Zusammenfassung

- Mächtiger als reguläre Sprachen
 - Regulär $\subset$ VPL $\subset$ CFL
- Eigenschaften von regulären Sprachen
 - Abgeschlossen unter Schnitt, Vereinigung, Komplement
- Determinisierung möglich

Quellen

[1] <http://www.informatik.uni-bremen.de/theorie/teach/lehre/thi1/WS1011/Folien/KA.pdf>
(09.12.2016)

[2] R. Alur, P. Madhusudan. Visibly Pushdown Languages. In
Proceeding
STOC '04 Proceedings of the thirty-sixth annual ACM symposium on
Theory of computing.