1711

Formalisierten Darstellung des *APOS*-Verfahrens zur datengesteuerten Programmierung

Markus Lepper

Vorläufige Version, 11. Juli 1991

Inhaltsverzeichnis

1 Gegenstand.

Das *APOS* genannte Verfahren zur Realisierung von datengesteuerter Programmierung wurde vom Autor in den achtziger Jahren entwickelt.

Im Rahmen eines Forschungsprojektes der Firma *micro-control* wurden unter Mitarbeit des Hrn. Th. Neuhaus mehrere Implementierungen von *APOS* entwickelt, getestet und angewandt, zunächst als Betriebssoftware für die AUDIAC-Serie der Firma *micro-control* unter besonderer Berücksichtigung der Anwendung auf dem Gebiet der Musik.

Von den zentralen Eigenschaften der *APOS*-Programmiersprache seien hier nur genannt ...

- minimaler Umfang des eigentlichen Sprachkernes,
- maximale Kombinabilität der Kontrollstrukturen,
- fast unbegrenzte Erweiterbarkeit und Redefinierbarkeit aller Systemressourcen.

Die von der Firma *micro-control* vorgenommenen Anwendungen beziehen sich in erster Linie auf die Programmierung und Koordinierung eines Verbundes von durchaus unterschiedlichen Signalprozessoren durch automatische Übersetzung von abstrakt definierten, den vom Anwender gewünschten Datenfluß beschreibenden Netzwerken.

APOS eignet sich unserer Erfahrung nach besonders für Anwendungen im wissenschaftlichen Bereich, die Verwaltung und Manipulation von Daten mit fein differenzierter Struktur durch isomorphe Methoden, die Entwicklung von Mensch-Maschine-Schnittstellen, als Grundlage für ein Betriebssystem für eine persönliche Datenbasis auf sog. workstations, die Konfigurierung von Realzeit-Prozeßsystemen mit grafischen Mitteln et.c.

Die von der Firma *micro-control* vorgenommenen Implementierungen bezogen sich der Reihe nach auf die Mikroprozessoren "80186", "80386" und "80486". Letztere Version befindet sich z.Zt. im Test in der Folkwang-Hochschule in Essen-Werden. Auf diese Implementierung werden wir im Text als "*A4*" referieren.

2 Übersicht der folgenden Kapitel.

In den nächsten drei Kapiteln werden die zentralen Bestandteile der Definition von *APOS* dargestellt. Diese sind ...

1. der Blockauswertungsmechanismus,
2. das Methodenkonzept,
3. die minimal notwendige *APOS*–Datenbasis der Schale 0.

3 Der Blockauswertungsmechanismus

Die Darstellung des *APOS*-internen Blockauswertungsmechanismus geschieht in Form von Übergangsregeln für einen deterministischen endlichen Automaten. Es werden zunächst die grundlegenden, auch in Kapitel 4 und 5 benötigten Mengen eingeführt.

Die Darstellungsform ähnelt sehr der Implementierung *A4*. Die genauen Definitionen beider weichen jedoch in einigen Einzelheiten von einander ab.

Diese Darstellung kann durchaus einen wichtigen Zweck erfüllen bei eventuellen Reimplementierungen, da sie auf einem vereinfachten und aufs Wesentliche beschränkten Niveau einige Zusammenhänge deutlicher darstellt, als die auf Laufzeitoptimalität hin entworfene Codierung in *A4*. *A4* liegt seinerseits in zwei Formen vor: als Pascal– und als Assembler–Version.

Diese drei Automaten (die beiden *A4*–Versionen und die vorliegende Darstellung) verhalten sich nicht völlig identisch. In der vorliegenden Darstellung werden z.B. nicht erforschte Situationen nicht dargestellt, wie z.B. $\langle \overline{cont} \cdot a_1 \cdot \overline{semi} \rangle$ oder $\langle \alpha_1 \cdot (\cdot \overline{cont} \ldots \rangle$!

Nicht dargestellt sind weiterhin der „TempMem"-Mechanismus und die "*if*"-Optimierung.

3.1 Anmerkungen und Schreibweisen.

Mengen : Wir verwenden für einige Mengen folgende Namen :

$$\begin{aligned} \mathsf{N} &= \{1,2,3,\ldots\} \\ \mathsf{N}_0 &= \{0,1,2,\ldots\} \\ \mathsf{D}_f &= \text{Definitionsmenge der Funktion } f \\ \mathsf{W}_f &= \text{Wertemenge der Funktion } f \end{aligned}$$

Relationen : Die n-fache Anwendung einer Relation bezeichnen wir mit :

$$\begin{aligned} R^0 &= \{(a,a) \mid \exists b \text{ mit } (a,b) \in R\} \\ R^1 &= R \\ R^n &= \{(a,b) \mid (a,c) \in R^{n-1} \wedge (c,b) \in R\} \\ R^+ &= \bigcup_{k=1}^{n} R^k \qquad n \to \infty \\ R^* &= R^0 \cup R^+ \end{aligned}$$

Funktionen : Die n-fache Anwendung einer Funktion bezeichnen wir mit :

$$\begin{aligned} f^0(q) &= q \\ f^1(q) &= f(q) \\ f^n(q) &= f(f^{n-1}(q)\,) \\ f^+(q) &= \bigcup_{k=1}^{n} \{f^k(q)\} \qquad n \to \infty \\ f^*(q) &= \{f^0(q)\} \cup f^+(q) \end{aligned}$$

NB:

$f^{n}(q) \in W_f, f^{+}(q), f^{*}(q) \subset W_f$!

n-tupel: Für Elemente des Kreuzproduktes mehrerer Mengen führen wir folgende Schreibweisen ein:

1. n-tupel bezeichnen wir i.A. mit kleinen griechischen Buchstaben.
2. ε bezeichnet die leere Liste () (quasi 0-tupel).
3. Der Sternoperator (*) bezeichnet die Menge aller n-tupel.
4. Es gilt für beliebige Mengen M:

$$\begin{aligned} M^0 &= \{\varepsilon\} \\ M^1 &= M \\ M^n &= M^{n-1} \times M \qquad \text{für alle } n \in N \\ M^+ &= \bigcup_{k=1}^{n} M^k \qquad n \to \infty \\ M^* &= M^+ \cup M^0 \end{aligned}$$

5. Ein $\mu \in M^n$ kann auf zwei Weisen geschrieben werden:

$$\mu = (a_1, a_2, \ldots, a_n) \quad \text{oder} \quad \mu = a_1 \cdot a_2 \cdot \ldots \cdot n$$

6. für zwei n-tupel $\mu_1 = (a_1, \ldots, a_n), \mu_2 = (b_1, \ldots, b_n) \in M^*$ gilt:

$$\begin{aligned} \mu_1 \cdot \mu_2 &\stackrel{def}{=} (a_1, \ldots, a_n, b_1, \ldots, b_n) \\ \mu_1 \cdot \varepsilon &\stackrel{def}{=} \mu_1 \\ \varepsilon \cdot \mu_1 &\stackrel{def}{=} \mu_1 \end{aligned}$$

7. Entsprechend gilt für $m \in M$ und $\mu = (a_1, \ldots, a_n) \in M^*$

$$m \cdot \mu \stackrel{def}{=} (m, a_1, \ldots, a_n) \quad \text{und} \quad \mu \cdot m \stackrel{def}{=} (a_1, \ldots, a_n, m)$$

8. Wir definieren eine Funktion, welche die Länge eines n-Tupels liefert :

$$\text{len} : M^* \to N; \qquad \text{len}(\mu) \stackrel{def}{=} \begin{cases} (k+1) & \text{für } \mu = (a_0, a_1, \ldots, a_k) \in M^+ \\ 0 & \text{für } \mu = \varepsilon \end{cases}$$

9. Wir definieren den Zugriff auf die Komponenten eines n-Tupels durch inidizierte Schreibweise :

Für $\mu = (a_0, a_1, \ldots, a_{(len(\mu)-1)}) \in M^+$ und $0 \leq k < \text{len}(\mu)$ gilt :
$\mu[k] \stackrel{def}{=} a_k$

3.2 Definitionen grundlegender *APOS*-eigener Mengen.

(1) Ω_t = Menge aller zum Zeitpunkt t existierenden Objekte

(2) Γ_t = Menge aller zum Zeitpunkt t existierenden Klassen mit $\Omega_t \cap \Gamma_t = \{\}$

(3) $c_0 \in \Gamma_t$ für alle t.

(4)
$$\Psi = \{f | \mathsf{D}_f = \Gamma_t \setminus \{c_0\}, \mathsf{W}_f \subseteq \Gamma_t, c_n \notin f^+(c_n) \land c_0 \in f^+(c_n) \text{ für alle } c_n \in \Gamma_t \setminus \{c_0\}\}$$

superClass steht für ein beliebiges $f \in \Psi$

(5) $\text{basisKlasse} = \Omega_t \to \Gamma_t$

Für jedes $q \in \Omega_t$ existiert basisKlasse(q)

(6) $O_{ref} = \{(q, c) | \; q \in \Omega_t \land c \in \text{superClass}^*(\text{basisKlasse}(q))\} \subset \Omega_t \times \Gamma_t$

Projektionsfunktionen:

$$\begin{array}{llll} \text{ident} & : O_{ref} \to \Omega_t; & \text{ident}(q, c) & = q \\ \text{class} & : O_{ref} \to \Gamma_t; & \text{class}(q, c) & = c \end{array}$$

(7)
$$\begin{array}{lcl} \mathit{QuoteModes} & = & \{\mathit{quoteSuper}, \mathit{quoteSame}, \mathit{quote0}, \mathit{quoteRest}\} \\ \mathit{Quotes} & = & (\{\mathit{quoteSuper}, \mathit{quoteSame}\} \times \{0, 1\} \times \mathsf{N}_0) \\ & \cup & (\{\mathit{quote0}, \mathit{quoteRest}\} \times \mathsf{N}_0 \times \mathsf{N}_0) \end{array}$$

mit $\mathit{QuoteModes} \cap \Gamma_t = \{\}$ und $\mathit{QuoteModes} \cap \Omega_t = \{\}$

Projektionsfunktionen:

$$\begin{array}{llll} \text{mode} & : \mathit{Quotes} \to \mathit{QuoteModes}; & \text{mode}(m, l, n) & = m \\ \text{level} & : \mathit{Quotes} \to \mathsf{N}_0; & \text{level}(m, l, n) & = l \\ \text{number} & : \mathit{Quotes} \to \mathsf{N}_0; & \text{number}(m, l, n) & = n \end{array}$$

(8) $\mathit{Conts} = (\{\mathit{cont}\} \times \mathsf{N})$
mit $\mathit{cont} \notin (\Omega_t \cup \Gamma_t)$

(9) $X_{ref} = O_{ref} \cup \mathit{Quotes} \cup \mathit{Conts}$

(10)
$$\begin{array}{lcl} B_0 & = & X_{ref}^+ \\ B_1 & = & (X_{ref} \cup B_0)^+ \\ B_2 & = & (X_{ref} \cup B_0 \cup B_1)^+ \\ B_n & = & (X_{ref} \cup (\bigcup_{k=0}^{(n-1)} B_k))^+ \\ B & = & \bigcup_{k=0}^{n} B_k \quad n \to \infty \\ B_{ref} & = & \{\mathit{kcBlock}\} \times B, \quad \text{mit } \mathit{kcBlock} \in \Gamma_t \text{ für alle } t. \end{array}$$

Projektion : mit $b = (kcBlock, \beta) \in B_{ref}$ gilt:

$$\text{ident}(b) \stackrel{def}{=} \beta$$

(11) $Y_{ref} = X_{ref} \cup B_{ref}$

$$\begin{array}{llll}
 & c_{tail} & \notin & (\Gamma_t \cup \Omega_t) \text{ für alle } t \\
(12) & \tilde{\Gamma}_t^+ & \stackrel{def}{=} & \Gamma_t^+ \times \{\varepsilon, c_{tail}\} \quad \text{für alle } \Gamma_t. \\
 & M_t & \subset & \tilde{\Gamma}_t^+ \times B \quad \text{für alle } t,
\end{array}$$

Dabei muß für alle $(\gamma_1, \beta_1), (\gamma_2, \beta_2) \in M_t$, mit $\gamma_1, \gamma_2 \in \tilde{\Gamma}_t^+$ und $\beta_1, \beta_2 \in B$ gelten : $(\gamma_1 = \gamma_2) \Rightarrow (\beta_1 = \beta_2)$.

#74

Projektionen : mit $m = (\alpha, \beta) \in M_t$ und $\alpha \in \tilde{\Gamma}_t^+$, $\beta \in B$ gilt:

$$\begin{array}{lll}
\text{defPat}(m) & \stackrel{def}{=} & \alpha \\
\text{defClass}(m, i) & \stackrel{def}{=} & \alpha[i] \quad \text{für } 0 \le i < len(\alpha) \\
\text{realisation}(m) & \stackrel{def}{=} & \beta
\end{array}$$

$$(13) \quad \begin{array}{lll}
Frame & = & Y_{ref}^* \times Y_{ref}^* \\
 & & \times (M_t \cup \{M_{null}\}) \\
 & & \times \{autoOn, autoOff\} \\
 & & \times \{quoteOn, quoteOff\}
\end{array}$$

Dabei gilt $\{autoOn, autoOff, quoteOn, quoteOff, M_{null}\} \cap (\Gamma_t \cup \Omega_t) = \{\}$

Projektionen : mit $F = \langle \alpha_1, \beta_1, M, a, q \rangle \in Frame$ gilt:

$$\begin{array}{lll}
\text{stack}(F) & \stackrel{def}{=} & \alpha_1 \\
\text{code}(F) & \stackrel{def}{=} & \beta_1 \\
\text{methode}(F) & \stackrel{def}{=} & M \\
\text{backMode}(F) & \stackrel{def}{=} & a \\
\text{quoteMode}(F) & \stackrel{def}{=} & q
\end{array}$$

(14) $IpretState = (Frame^+) \cup FEHLERFALL.$

(15) $\vdash\ = IpretState \mapsto IpretState$ ist links-totale Funktion.

Satz 1 *Mit* $f_0 = (\langle \varepsilon, \beta, M_{null}, autoOn, quoteOff \rangle) \in IpretState$
und $\beta \in B$ *und* $f_1 = (\langle \varepsilon, \varepsilon, M_{null}, autoOn, quoteOff \rangle)$ *gilt :*

$f_0 \vdash^{n} f_1$ *für genügend großes* n . Bew.

(16) $\{with, if, back, splice, ok, quote, semi\} \subset \Omega_t$ für alle t.

für $q \in \{with, if, back, splice, ok, quote, semi\}$ schreiben wir :

$$\overline{q} = (q, \text{basisKlasse}(q)) \in O_{ref}$$

(17)
$$\{\text{Prec}, \text{Rrec}\} \cap (\Omega_t \cup \Gamma_t) = \{\} \qquad \text{für alle } t.$$
$$\text{frameTyp} : Frame \rightarrow \{\text{Rrec}, \text{Prec}\};$$
$$\text{frameTyp}(f) \overset{def}{=} \begin{cases} \text{Rrec} & \text{für } ((code(f) = \varepsilon) \vee (code(f)[0] = \overline{semi})) \\ \text{Prec} & \text{für } ((code(f) \neq \varepsilon) \wedge (code(f)[0] \neq \overline{semi})) \end{cases}$$

(18)
$$\text{lastRrec} : Frame \rightarrow \mathsf{N}_0 \cup \{-1\};$$
$$\text{lastRrec}(F_{n,t}) \overset{def}{=} \begin{cases} k & \text{wenn es ein } k \text{ gibt mit } 0 \leq k < n \\ & \text{und frameTyp}(F_{k,t}) = \text{Rrec} \\ & \text{und frameTyp}(F_{m,t}) \neq \text{Rrec} \\ & \text{für alle } m \text{ mit } k < m < n\ . \\ -1 & \text{andernfalls.} \end{cases}$$

(19)
$$\text{lastPrec} : Frame \rightarrow \mathsf{N}_0 \cup \{-1\};$$
$$\text{lastPrec}(F_{n,t}) \overset{def}{=} \begin{cases} k & \text{wenn es ein } k \text{ gibt mit } 0 \leq k < n \\ & \text{und frameTyp}(F_{k,t}) = \text{Prec} \\ & \text{und frameTyp}(F_{m,t}) \neq \text{Prec} \\ & \text{für alle } m \text{ mit } k < m < n\ . \\ -1 & \text{andernfalls.} \end{cases}$$

3.3 Anmerkungen zur Schreibweise der Übergangsregeln.

Die Übergangsfunktion $\vdash$ operiert auf *IpretState* . Wir stellen die Komponenten jedes $i = (F_0, F_1, \ldots, F_n) \in IpretState$ von oben nach unten dar.

Die Auslassungszeichen ... haben je Ort eine andere Bedeutung:

1. Zwischen den *Frames*: Wert der *Frames* geht nicht in die Eingangsbestimmung der Übergangsvorschrift ein. Hier können 0 bis n *Frames* anstelle der ... stehen.

2. Innerhalb eines *Frames* auf der linken Seite: Wert des Feldes geht nicht in die Eingangsbestimmung der Übergangsregel ein.

3. Innerhalb eines *Frames* auf der rechten Seite, der vor Regelanwendung schon existiert: Wert dieses Feldes ändert sich durch Anwenden der Regel nicht.

4. Innerhalb eines neuen *Frames* F_n, der vor der Regelanwendung nicht existierte: Wert dieses Feldes ist identisch mit dem Wert des selben Feldes im vorhergehenden *Frame* F_{n-1}.

Ein eingetragener Wert auf der linken Seite einer Regel bedeutet, daß dieser Wert vorliegen *muß*, damit die Regel angewandt werden darf.

Ein eingetragener Wert auf der rechten Seite einer Regel bedeutet, daß nach Regelanwendung der *IpretState* sich entsprechend verändert.

Der Index "t" auf der linken Seite der Übergangsvorschrift und der Index "t+1" auf der rechten Seite sind i.A. abkürzenderweise nicht ausgeschrieben.

$$
\begin{array}{llllll|llllll}
F_{0,t}\langle & & \ldots & & & \rangle & F_{0,t+1} & \langle & & \ldots & & \rangle \\
\ldots & & (2) & & & & & & & (3) & & \\
(1) & & & & & & & & & & & \\
 & & & & & \vdash & & & & & & \\
\ldots & & & & & & & & & & & \\
F_{n,t}\langle & \ldots, & \overline{semi}\cdot\alpha_1, & & \ldots & \rangle & F_{n,t+1} & \langle & \ldots, & \alpha_1, & \ldots & \rangle \\
 & (2) & & & & & & & (3) & & & \\
 & & & & & & F_{n+1,t+1} & \langle & \ldots, & \mu, & \ldots & \rangle \\
 & & & & & & & & (4) & & &
\end{array}
$$

3.4 Definition der einzelnen Übergangsregeln.

Vorbemerkung :
Wenn nicht anders angegeben, gilt im Folgenden
$\alpha_1, \alpha_2, \beta_1, \beta_2 \in Y^*_{ref}$ $\alpha, \beta \ldots \in Y_{ref}$
und $n \in \mathsf{N}_0$.

3.4.1 Einfache Objektreferenzen.

$$(20) \quad \left| \, F_n \quad \langle \alpha_1, \quad a \cdot \alpha_2, \quad \ldots \rangle \, \right| \quad \vdash \quad \left| \, F_n \quad \langle \alpha_1 \cdot a, \quad \alpha_2, \quad \ldots \rangle \, \right|$$

für $a \in (O_{ref} \setminus \{\overline{semi}, \overline{quote}\}) \cup Conts$.

3.4.2 Erzeugen von Blockobjekten, der *quote*-Operator.

$$(21) \quad \left| \, F_n \quad \langle \epsilon, \overline{quote} \cdot \alpha_1, \ldots, quoteOff \rangle \, \right| \quad \vdash \quad \left| \, F_n \quad \langle \epsilon, \alpha_1, \ldots, quoteOn \rangle \, \right|$$

für $\alpha_1 \in Y^*_{ref}$.

$$(22) \quad \left| \, F_n \quad \langle \epsilon, \overline{quote} \cdot \alpha_1, \ldots, quoteOn \rangle \, \right| \quad \vdash \quad \left| \, F_n \quad \langle \overline{quote}, \alpha_1, \ldots \rangle \, \right|$$

für $\alpha_1 \in Y^*_{ref}$.

$$(23) \quad \left| F_n \quad \langle \alpha_2, \overline{quote} \cdot \alpha_1, \ldots \rangle \right| \quad \vdash \quad \left| F_n \quad \langle \alpha_2 \cdot \overline{quote}, \alpha_1, \ldots \rangle \right|$$

für $\alpha_1 \in Y^*_{ref}, \alpha_2 \in Y^+_{ref}$.

$$(24) \quad \left| F_n \quad \langle \alpha_1, \varepsilon, \ldots, quoteOn \rangle \right| \quad \vdash \quad \left| F_n \quad \langle \overline{back} \cdot b, \varepsilon, \ldots, quoteOff \rangle \right|$$

mit $b = (kcBlock, \alpha_1) \in B_{ref}$.

$$(25) \quad \left| F_n \quad \langle \alpha_1, \overline{semi} \cdot \alpha_2, \ldots quoteOn \rangle \right| \quad \vdash \quad \left| F_n \quad \langle \alpha_1 \cdot \overline{semi}, \alpha_2, \ldots \rangle \right|$$

3.4.3 Schachteln von Blöcken, P-Rekursion.

$$(26) \quad \left| F_n \quad \langle \alpha_1, b \cdot \alpha_2, \ldots \rangle \right| \quad \vdash \quad \left| \begin{array}{llllll} F_n & \langle \alpha_1 \cdot \overline{ok}, & b \cdot \alpha_2, & \ldots, & autoOn & \ldots \quad \rangle \\ F_{n+1} & \langle \varepsilon, & \hat{b}, & M_{null}, & autoOn, & quoteOff \quad \rangle \end{array} \right|$$

[illegible]!?

mit $b \in B_{ref}$ und $\hat{b} = \mathrm{ident}(b)$.

3.4.4 Anwenden von Methoden, R-Rekursion.

Vorbemerkung :
Die Definition der Funktion methFind(M_t, β_1) ist Gegenstand des nächsten Abschnittes (4.2).

$$(27) \quad \left| F_n \quad \langle \alpha_1, \quad \varepsilon, \quad \ldots, r, quoteOff \quad \rangle \right| \quad \vdash \quad \left| \begin{array}{llll} F_n & \langle \alpha_1, & \ldots, & M, \ldots \quad \rangle \\ F_{n+1} & \langle \varepsilon, & \mu, & M_{null}, r, quoteOff \quad \rangle \end{array} \right|$$

d!!

mit $r \in \{autoOn, autoOff\}$
und $\alpha_1 \neq \varepsilon, \qquad \alpha_1[0] \notin \{with, splice, back\} \cup Conts$
und $M \in$ methFind(M_t, α_1) und $\mu =$ realisation(M).

$$(28) \quad \left| F_n \quad \langle \alpha_1, \varepsilon, \ldots, quoteOff \rangle \right| \quad \vdash \quad \left| \text{Error : no Methode} \right|$$

mit α_1 wie in (27) und methFind(M_t, α_1) = {}.

$$(29)\quad \left| F_n \quad \langle \alpha_1, \overline{semi}\cdot\alpha_2, \ldots, r, quoteOff\rangle \right|$$

? korrekt!

$$\vdash \left| \begin{array}{lllll} F_n & \langle\alpha_1, & \overline{semi}\cdot\alpha_2, & M, & \ldots\rangle \\ F_{n+1} & \langle\varepsilon, & \mu, & M_{null}, & r, quoteOff\rangle \end{array} \right|$$

mit r, α_1, M und μ wie in (27)

$$(30)\quad \left| F_n \quad \langle \alpha_1, \overline{semi}\cdot\alpha_2, \ldots, quoteOff\rangle \right| \vdash \left| \textsf{Error: no Methode} \right|$$

mit α_1 wie in (27) und methFind$(M_t, \alpha_1) = \{\}$.

3.4.5 Pseudomethoden.

$$(31)\quad \left| \begin{array}{llll} F_{n-1} & \langle\alpha_1\cdot\overline{ok}, & \beta_{n-1}, & \ldots\rangle \\ F_n & \langle\overline{splice}\cdot b, & \varepsilon, & \ldots\rangle \end{array} \right| \vdash \left| F_{n-1} \quad \langle\alpha_1\cdot\hat{b}, \quad \beta_{n-1}, \quad \ldots\rangle \right|$$

Warum β_{n-1}? und nicht einfach α_2

mit $b \in B_{ref}, \hat{b} = \mathrm{ident}(b)$ und $(n-1) = \mathrm{lastPrec}(F_n)$.

$$(32)\quad \left| F_n \quad \langle\overline{with}\cdot\alpha_1\cdot b, \quad \varepsilon, \quad \ldots\rangle \right| \vdash \left| \begin{array}{lllll} F_n & \langle\alpha_1, & \varepsilon, & M_{null}, & \ldots\rangle \\ F_{n+1} & \langle\varepsilon, & \hat{b}, & M_{null}, & autoOn, quoteOff\rangle \end{array} \right|$$

mit $b \in B_{ref}$, $\hat{b} = \mathrm{ident}(b)$ und $\alpha_1 \in Y^+_{ref}$.

$$(33)\quad \left| F_n \quad \langle\overline{with}\cdot\alpha_1\cdot b, \overline{semi}\cdot\alpha_2, \ldots\rangle \right|$$

$$\vdash \left| \begin{array}{lllll} F_n & \langle\alpha_1, & \overline{semi}\cdot\alpha_2, & M_{null}, & \ldots\rangle \\ F_{n+1} & \langle\varepsilon, & \hat{b}, & M_{null}, & autoOn, quoteOff\rangle \end{array} \right|$$

mit $b \in B_{ref}$, $\hat{b} = \mathrm{ident}(b)$ und $\alpha_1, \alpha_2 \in Y^+_{ref}$.

besser: "n" nicht in Def

3.4.6 Zugriff auf Nachrichtenkomponenten.

$$(34)\quad \left| F_n \quad \langle\alpha_1, \quad (q, l, h)\cdot\alpha_2, \quad \ldots\rangle \right| \vdash \left| F_n \quad \langle\alpha_1\cdot(q, (l-1), h), \quad \alpha_2, \quad \ldots\rangle \right|$$

mit $q \in QuoteModes, l \geq 1$. $n \in \mathbb{N}^0$

$$(35)\quad \left| \begin{array}{lll} F_k & \langle\beta_1, & \ldots, \quad \ldots\rangle \\ & & \\ F_n & \langle\alpha_1, & (quote0, 0, h)\cdot\alpha_2, \quad \ldots\rangle \end{array} \right| \vdash \left| \begin{array}{l} \ldots \\ \ldots \\ F_n \quad \langle\alpha_1\cdot\beta_1[h], \quad \alpha_2, \quad \ldots\rangle \end{array} \right|$$

mit $k = \mathrm{lastRrec}(F_n) > 0$ und $0 \leq \overset{n}{\cancel{h}} < \mathrm{len}(\beta_1)$.   T

$$(36)\quad \left|\begin{array}{lllll} F_k & \langle\beta_1, & \ldots & M, & \ldots\rangle \\ \ldots & & & & \\ F_n & \langle\alpha_1, & (q,0,h)\cdot\alpha_2, & \ldots, & \ldots\rangle \end{array}\right| \vdash \left|\begin{array}{lllll} \ldots & & & & \\ \ldots & & & & \\ F_n & \langle\alpha_1\cdot a, & \alpha_2, & \ldots, & \ldots\rangle \end{array}\right|$$

für $q \in \{\mathit{quoteSame}, \mathit{quoteSuper}\} \subset \mathit{QuoteModes}$ und
$k = \mathrm{lastRrec}(F_n) > 0$ und $0 \leq h < \mathrm{len}(\beta_1)$.

Dabei gilt

$$a = \begin{cases} (\mathrm{ident}(\beta_1[h]), \mathrm{defClass}(M,h)) & \text{für } q = \mathit{quoteSame} \\ (\mathrm{ident}(\beta_1[h]), \mathrm{superClass}(\mathit{defclass}(M,h))) & \text{für } q = \mathit{quoteSuper} \end{cases}$$

$$(37)\quad \left|\begin{array}{llll} F_k & \langle\beta_1, & \ldots, & \ldots\rangle \\ \ldots & & & \\ F_n & \langle\alpha_1, & (q,0,h)\cdot\alpha_2, & \ldots\rangle \end{array}\right| \vdash \left|\begin{array}{l} \ldots \\ \ldots \\ \mathsf{ERROR : no\ AccNum} \end{array}\right|$$

für $q \in \{\mathit{quote0}, \mathit{quoteSame}, \mathit{quoteSuper}\} \subset \mathit{QuoteModes}$ und
$k = \mathrm{lastRrec}(F_n) > 0$ und $h \geq \mathrm{len}(\beta_1)$.

$$(38)\quad \left|\begin{array}{llll} F_n & \langle\alpha_1, & (q,0,h)\cdot\alpha_2, & \ldots\rangle \end{array}\right| \vdash \left|\begin{array}{l} \mathsf{ERROR : no\ such\ Level.} \end{array}\right|$$ A4 ??

für $q \in \mathit{QuoteModes}$ und $k = \mathrm{lastRrec}(F_n) < 0$.

$$(39)\quad \left|\begin{array}{llll} F_k & \langle\beta_1, & \ldots, & \ldots\rangle \\ \ldots & & & \\ F_n & \langle\alpha_1, & (\mathit{quoteRest},0,h)\cdot\alpha_2, & \ldots\rangle \end{array}\right| \vdash \left|\begin{array}{llll} \ldots & & & \\ \ldots & & & \\ F_n & \langle\alpha_1\cdot\gamma, & \alpha_2, & \ldots\rangle \end{array}\right|$$

für $k = \mathrm{lastRrec}(F_n)$.

Dabei gilt mit $l = \mathrm{len}(\beta_1)$:

$$\gamma = \begin{cases} (\beta_1[h]\cdot\beta_1[h+1]\ldots\cdot\beta_1[l-1] & \text{für } h < l \\ \varepsilon & \text{für } h \geq l \end{cases}$$

3.4.7 Ende der Methodenausführung.

$$(40)\quad \left|\begin{array}{llll} F_{n-1} & \langle\alpha_1, & \varepsilon, & \ldots\rangle \\ F_n & \langle\varepsilon, & \varepsilon, & \ldots\rangle \end{array}\right| \;\vdash\; \left|\begin{array}{llll} F_{n-1} & \langle\varepsilon, & \varepsilon, & \ldots\rangle \end{array}\right|$$

mit $n > 0$ und $\alpha_1 \in Y^+_{ref}$.

$$(41)\quad \left|\begin{array}{llll} F_{n-1} & \langle\alpha_1, & \overline{semi}\cdot\alpha_2, & \ldots\rangle \\ F_n & \langle\varepsilon, & \varepsilon, & \ldots\rangle \end{array}\right| \;\vdash\; \left|\begin{array}{llll} F_{n-1} & \langle\varepsilon, & \alpha_2, & \ldots\rangle \end{array}\right|$$

mit $\alpha_1, \alpha_2 \in Y^+_{ref}$.

$$(42)\quad \left|\begin{array}{llll} F_{n-1} & \langle\alpha_1, & b\cdot\alpha_2, & \ldots\rangle \\ F_n & \langle\varepsilon, & \varepsilon, & \ldots\rangle \end{array}\right| \;\vdash\; \left|\begin{array}{llll} F_{n-1} & \langle\alpha_1, & \alpha_2, & \ldots\rangle \end{array}\right|$$

Ende Proc

mit $\alpha_1, \alpha_2 \in Y^*_{ref}$ und $b \in B_{ref}$.

3.4.8 Der *back*-Operator.

F_k

$m-1$

$$(43)\quad \left|\begin{array}{lllll} F_m & \langle\alpha_m\cdot x_m, & \ldots, & \ldots, & \ldots\rangle \\ \ldots & & & & \\ F_{n-2} & \langle\ldots, & \ldots, & autoOn, & \ldots\rangle \\ F_{n-1} & \langle\ldots, & \ldots, & \ldots, & \ldots\rangle \\ F_n & \langle\overline{back}\cdot a, & \varepsilon, & \ldots, & \ldots\rangle \end{array}\right| \;\vdash\; \left|\begin{array}{lllll} F_m & \langle\alpha_m\cdot a, & \ldots, & \ldots, & \ldots\rangle \\ \ldots & & & & \\ F_{n-2} & \langle\ldots, & \ldots, & \ldots, & \ldots\rangle \\ F_{n-1} & \langle\ldots, & \ldots, & autoOff, & \ldots\rangle \\ F_n & \langle\varepsilon, & \varepsilon, & \ldots, & \ldots\rangle \end{array}\right|$$

mit $(m-1) = \text{lastRrec}(F_n)$ und $m = \text{lastPrec}(n-1) \leq (n-2)$.

$$(44)\quad \left|\begin{array}{lllll} F_m & \langle\alpha_m\cdot x_m, & \ldots, & \ldots, & \ldots\rangle \\ \ldots & & & & \\ F_{n-2} & \langle\ldots, & \ldots, & autoOff, & \ldots\rangle \\ F_{n-1} & \langle\ldots, & \ldots, & \ldots, & \ldots\rangle \\ F_n & \langle\overline{back}\cdot a, & \varepsilon, & \ldots, & \ldots\rangle \end{array}\right| \;\vdash\; \left|\begin{array}{lllll} F_m & \langle\alpha_m\cdot x_m, & \ldots, & \ldots, & \ldots\rangle \\ \ldots & & & & \\ F_{n-2} & \langle\ldots, & \ldots, & \ldots, & \ldots\rangle \\ F_{n-1} & \langle\ldots, & \ldots, & autoOff, & \ldots\rangle \\ F_n & \langle\varepsilon, & \varepsilon, & \ldots, & \ldots\rangle \end{array}\right|$$

mit $(n-1) = \text{lastRrec}(F_n)$ und $m = \text{lastPrec}(n-1) \leq (n-2)$.

$$(45)\quad \left|\begin{array}{lllll} F_k & \langle\alpha_k\cdot x_k, & \ldots, & \ldots, & \ldots\rangle \\ \ldots & & & & \\ F_{m-1} & \langle\ldots, & \ldots, & autoOn, & \ldots\rangle \\ F_m & \langle\ldots, & \ldots, & \ldots, & \ldots\rangle \\ \ldots & & & & \\ F_{n-1} & \langle\beta_1\cdot x_n, & \ldots, & \ldots, & \ldots\rangle \\ F_n & \langle\overline{back}\cdot a, & \varepsilon, & \ldots, & \ldots\rangle \end{array}\right| \;\vdash\; \left|\begin{array}{lllll} F_k & \langle\alpha_k\cdot a, & \ldots, & \ldots, & \ldots\rangle \\ \ldots & & & & \\ F_{m-1} & \langle\ldots, & \ldots, & \ldots, & \ldots\rangle \\ F_m & \langle\ldots, & \ldots, & autoOff, & \ldots\rangle \\ \ldots & & & & \\ F_{n-1} & \langle\beta_1\cdot a, & \ldots, & \ldots, & \ldots\rangle \\ F_n & \langle\varepsilon, & \varepsilon, & \ldots, & \ldots\rangle \end{array}\right|$$

mit $m = \text{lastRrec}(F_n) < (n-1)$ und $k = \text{lastPrec}(m) \leq (m-1)$.

$$(46) \quad \left| \begin{array}{lllll} F_k & \langle \alpha_k \cdot x_k, & \ldots, & \ldots, & \ldots\rangle \\ \ldots & & & & \\ F_{m-1} & \langle \ldots, & \ldots, & autoOff, & \ldots\rangle \\ F_m & \langle \ldots, & \ldots, & \ldots, & \ldots\rangle \\ \ldots & & & & \\ F_{n-1} & \langle \beta_1 \cdot x_n, & \ldots, & \ldots, & \ldots\rangle \\ F_n & \langle \overline{back} \cdot a, & \varepsilon, & \ldots, & \ldots\rangle \end{array} \right| \vdash \left| \begin{array}{lllll} F_k & \langle \alpha_k \cdot x_k, & \ldots, & \ldots, & \ldots\rangle \\ \ldots & & & & \\ F_{m-1} & \langle \ldots, & \ldots, & \ldots, & \ldots\rangle \\ F_m & \langle \ldots, & \ldots, & autoOff, & \ldots\rangle \\ \ldots & & & & \\ F_{n-1} & \langle \beta_1 \cdot a, & \ldots, & \ldots, & \ldots\rangle \\ F_n & \langle \varepsilon, & \varepsilon, & \ldots, & \ldots\rangle \end{array} \right|$$

mit $m = \text{lastRrec}(F_n) < (n-1)$ und $k = \text{lastPrec}(m) \leq (m-1)$.

3.4.9 Die *cont*-Operatoren.

Mit Error $\notin (\Omega_t \cup \Gamma_t)$ für alle t gilt ...

$$lastCont : \text{Frame} \to (\text{Frame} \cup \text{Error})$$

$$(47) \quad lastCont(F_{n,t}) \stackrel{def}{=} \begin{cases} F_{k,t} & \text{wenn es ein } k \text{ gibt, mit } 0 \leq k < n \\ & \text{und code}(F_{k,t}) \neq \varepsilon \\ & \text{und code}(F_{m,t}) \not\ni \varepsilon \\ & \text{für alle } m \text{ mit } k < m < n. \\ \text{Error} & \text{wenn es kein solches } k \text{ gibt.} \end{cases}$$

≠ 0

⊢ nicht fix!!

Dann gilt ...

$$(48) \quad \left| \begin{array}{llll} F_k & \langle \beta_1, & \beta_2, & \ldots\rangle \\ \ldots & & & \\ F_{n-1} & \langle \ldots, & \ldots, & \ldots\rangle \\ F_n & \langle (cont, x) \cdot \alpha_1, & \varepsilon, & \ldots\rangle \end{array} \right| \vdash \left| \begin{array}{ll} F_k & \langle \alpha_1, \beta_2, \; M_{null}, autoOn, quoteOff\rangle \\ & \\ & \\ & \end{array} \right|$$

Mit $\alpha_1, \beta_1, \beta_2 \in Y^+_{ref}$,
und $x > 0$
und $lastCont^y(F_n) \neq$ Error für alle y mit $0 < y \leq x$
und $lastCont^x(F_n) = F_k$.

Prodes??

Back – Bug/Feature

$$(49) \quad \left| \; F_n \quad \langle (cont, x) \cdot \alpha_1, \quad \varepsilon, \quad \ldots\rangle \; \right| \vdash \left| \; \text{Error : no such Level.} \; \right|$$

Mit $x > 0$
und $lastCont^y(F_n) =$ Error für ein y mit $0 < y \leq x$.

3.4.10 Prozessor-Schnittstelle.

Zum Zweck der Definition einer Schnittstelle zum zugrunde liegenden Hardware-Prozessor erweitern wir die Menge "O_{ref}" wie folgt :

$$(50)\quad \begin{array}{lll} \{\$\mathsf{Push}, \$\mathsf{PushAcc}\} \times \mathsf{N}_0 & = \mathsf{PushCodes} & \subset O_{ref}, \\ \{\$\mathsf{CodeCall}\} \times \mathsf{N}_0 \times \mathsf{N}_0 \times \mathsf{N}_0 & = \mathsf{CodeCalls} & \subset O_{ref}, \end{array} \quad \ldots \text{ für alle } t\,.$$

class = $Push ...
ident = VVV

$$(51)\quad \left|\begin{array}{llll} F_{n-1} & \langle\beta_1, & \beta_2, & \ldots\rangle \\ F_n & \langle\varepsilon, & a\cdot\alpha_2 & \ldots\rangle \end{array}\right| \vdash \left|\begin{array}{llll} \ldots & & & \\ F_n & \langle\varepsilon, & \alpha_1, & \ldots\rangle \end{array}\right|$$

$\alpha_1 0$

mit $\beta_1 \in O^+_{ref}, \beta_2 \in Y^*_{ref}$,
und $\alpha_1 \in$ $\mathsf{PushCodes}^+ \times \mathsf{CodeCalls}$,
$\mathrm{lastRrec}(F_n) = n - 1$
$a = (a_0, x) \in \mathsf{PushCodes}$
und $0 \leq x < \mathrm{len}(\beta_1)$.

(*)

Beim Vollzug dieses Überganges wird die technische Repräsentation der Objektreferenz $\beta_1[x]$ auf den Parameter-Stack des Hardware-Prozessors abgelegt (qua "Push").

$$(52)\quad \left|\begin{array}{llll} F_{n-1} & \langle\beta_1, & \beta_2, & \ldots\rangle \\ F_n & \langle\varepsilon, & a\cdot\alpha_2, & \ldots\rangle \end{array}\right| \vdash \left|\ \mathsf{ERROR:\ no\ such\ AccNum.}\ \right|$$

mit $\beta_1, \beta_2, \alpha_1$ und $\mathrm{lastRrec}(F_n)$ wie in (51),
$a = (a_0, x) \in \mathsf{PushCodes}$
und $x \geq \mathrm{len}(\beta_1)$.

$$(53)\quad \left|\begin{array}{llll} F_{n-1} & \langle\beta_1, & \beta_2, & \ldots\rangle \\ F_n & \langle\varepsilon, & a, & \ldots\rangle \end{array}\right| \vdash \left|\begin{array}{ll} \ldots & \\ F_n & \langle\varepsilon, \varepsilon, \ \ldots\rangle \end{array}\right|$$

mit β_1, β_2 und $\mathrm{lastRrec}(F_n)$ wie in (51),
$a = (\$\mathsf{codeCall}, x, y, z) \in \mathsf{CodeCalls}$.

Während des Vollzuges dieses Überganges wird eine für die zugrunde liegenden Hardware entsprechend codierte externe Prozedur angesprungen. In der momentanen Implementierung "*A4*" bedeuten dabei x und y die Einsprungadresse ("Segment" und "Offset") des Code-Moduls und z die Adresse der Prozedur innerhalb des Moduls.

3.4.11 Randbedingungen.

Vorbemerkung : Die in den folgenden Regeln beschriebenen Übergänge haben jeweils eine Einwirkung eines "äußeren Interpretierers" zur Voraussetzung, also eine Eingabe seitens des "Benutzers" des inneren Interprertierers.

(54) $\left|\ \text{Error : < fehlerfall >}\ \right| \quad \vdash \quad \left|\ F_0 \quad \langle \varepsilon, \varepsilon, M_{null}, autoOn, quoteOff\rangle\ \right|$

Diese Übergang geschieht dann, wenn der Benutzer des Blockauswertungsmechanismus den Fehlerfall durch eine entsprechende Eingabe quittiert.

(55) $\left|\ F_{0,t} \quad \langle \varepsilon, \varepsilon, M_{null}, autoOn, quoteOff\rangle\ \right| \quad \vdash \quad \left|\ F_{0,t+1} \quad \langle \varepsilon, \mu, M_{null}, autoOn, quoteOff\rangle\ \right|$

Mit $\mu \in Y^{+}_{ref}$.

Diese Übergang geschieht dann, wenn der äußere Interpretierer eine Anfrage zwecks Auswertung an den Blockauswertungsmechanismus (= den inneren Interpretierer) übergibt. Das n-Tupel μ ist die interne Repräsentation dieser Anfrage.

4 Das Methodenkonzept.

4.1 Definition neuer Methoden.

Es gilt ...

$$
(56) \quad \begin{array}{lcl}
dM & \in & \Omega_t \text{ für alle } t. \\
\overline{dM} & \stackrel{def}{=} & (\text{basisKlasse}(dM), dm) \in O_{ref}. \\
tailRep, blockRep & \in & \Omega_t \text{ für alle } t.
\end{array}
$$

Mit

$$
cl : O_{ref} \to \Gamma_t; \qquad cl(o) \stackrel{def}{=} \begin{cases} c_{tail} & \text{wenn ident}(o) = tailRep \\ kcBlock & \text{wenn ident}(o) = blockRep \\ \text{class}(o) & \text{andernfalls.} \end{cases}
$$

und mit

$$
M_{t,\mu} \stackrel{def}{=} \{(\lambda,\nu) | (\lambda,\nu) \in M_t \land \lambda = \mu\} \subset M_t \qquad \text{für jedes } \mu \in \tilde{\Gamma}^{+}_t,
$$

und mit $\beta_2 \in B_{ref}$ und $\beta_1 \in ((O_{ref} \setminus \{tailRep\})^{+} \times \{\varepsilon, tailRep\})$ und $\gamma \in \tilde{\Gamma}^{+}_t$ mit $\text{len}(\gamma) = \text{len}(\beta)$ und $\gamma[i] = cl(\beta[i])$ für alle i mit $0 \le i < len(\beta)$

gilt dann ...

(57) $\left| F_{n,t} \quad \langle \overline{dM} \cdot \beta_1 \cdot \beta_2, \varepsilon, \ldots \rangle \right| \quad \vdash \quad \left| F_{n,(t+1)} \quad \langle \varepsilon, \varepsilon, \ldots \rangle \right|$

...wobei gilt :

$$M_{(t+1)} = M_t \setminus M_{t,\gamma_1} \cup \{(\gamma_1, \beta_2)\}$$

4.2 Bestimmen der anzuwendenden Methode.

(58)
$$\begin{array}{llll} M_{t,n} \stackrel{def}{=} & & \{m \in M_t \,| & (\mathrm{len}(\mathrm{defPat}(m)) = n) \\ & & & \wedge(\mathrm{defClass}(m, \mathrm{len}(m) - 1) \neq c_{tail})\} \\ & \cup & \{(\tilde{\gamma}, \beta)| & \mathrm{len}(\tilde{\gamma}) = n, \text{ und} \\ & & & \exists \bar{m} \in M_t, \text{ mit} \\ & & & (\mathrm{len}(\mathrm{defPat}(\bar{m})) \leq n) \\ & & & \wedge(\mathrm{defClass}(\bar{m}, \mathrm{len}(\bar{m}) - 1) = c_{tail}) \\ & & & \wedge(\mathrm{defClass}(\bar{m}, i) = \mathrm{defClass}(\tilde{\gamma}, i) \\ & & & \quad \text{für alle } i \text{ mit } 0 \leq i < \mathrm{len}(\bar{m}) - 1 \;) \\ & & & \wedge(\mathrm{defClass}(\tilde{\gamma}, j) = c_0 \\ & & & \quad \text{für alle } j \text{ mit } \mathrm{len}(\bar{m}) - 1 \leq j < n) \\ & & \}. & \end{array}$$

Mit $\mathrm{nv} \notin \Omega_t \cup \Gamma_t$ für alle t gilt ...

(59) $\delta : \Gamma_t \times \Gamma_t \rightarrow (N_0 \cup \{\mathrm{nv}\});$ $\qquad \delta(c_1, c_2) \stackrel{def}{=} \begin{cases} n, & \text{für superClass}^n(c_1) = c_2 \\ \mathrm{nv}, & \text{für superClass}^m(c_1) = c_2 \text{ für kein } m. \end{cases}$

Mit $\beta \in Y^+_{ref}, \mathrm{len}(\beta) = n$,
und $\mathrm{nv} > i$ für jedes $i \in \mathrm{N}_0$
gilt für jedes t :

(60)
$$\begin{array}{lll} \mathrm{methFind}(\mathrm{M_t}, \beta) \stackrel{def}{=} & \{(\mu, \nu) \in M_{t,n} | & \delta(\mathrm{class}(\beta[i]), \mu[i]) \geq 0, \\ & & \text{und für alle} (\mu', \nu') \in M_{t,n} \\ & & \text{und alle } i \text{ mit } 0 \leq i \leq n \text{ gilt :} \\ & & (\; \delta(\mathrm{class}(\beta[i]), \mu'[i]) < \delta(\mathrm{class}(\beta[i]), \mu[i]) \;) \\ & & \Rightarrow \exists j \text{ mit } 0 \leq j < i \\ & & \text{und } \delta(\mathrm{class}(\beta[j]), \mu'[j]) > \delta(\mathrm{class}(\beta[j]), \mu[j]). \\ & & \} \end{array}$$

Satz 2 *Mit* $\beta \in O^+_{ref}$ *und* $M_\beta = \mathrm{methFind}(\mathrm{M_t}, \beta)$ *gilt*
$CARD(M_\beta) = 0 \vee CARD(M_\beta) = 1.$

5 Die minimal notwendige Datenbasis der Schale 0