

DIE ERWEITERUNG DES LISPKIT LISP-SYSTEMS MIT
DEN ALLGEMEINEN FUNKTIONEN RPLACA UND RPLACD

Vojislav Stojković, Julijana Mirčevski
Prirodno-matematički fakultet. Institut za matematiku
21000 Novi Sad, ul.dr Ilije Djuričića br.4, Jugoslavija

ZUSAMMENFASSUNG

Im Beitrag werden die Befehle RPLACA und RPLACD der SECD-Maschine definiert.

Die eingeführten Befehle sind in der Gestalt von Programmbausteinen die in FORTRAN geschrieben werden, realisiert. Der Programmsimulator der SECD-Maschine wird um diese Bausteine erweitert.

Die gewöhnlichen, allgemeinen, pseudo- und ihnen entsprechenden wohlgeformten S-ausdrücke werden definiert.

Es werden auch die gewöhnlichen, allgemeinen und Pseudofunktionen definiert.

Die Sprache LISPKIT LISP wird durch die allgemeinen Funktionen RPLACA und RPLACD erweitert. Es wird darauf hingewiesen, warum die eingeführten Funktionen allgemein sind.

Die Übersetzung der allgemeinen Funktionen RPLACA und RPLACD in die Maschinsprache der SECD-Maschine wird definiert. Im Einklang mit ihr wird die entsprechende Modifikation des Compilers ausgeführt.

Das so erweiterte LISPKIT LISP-System wird in den Rechner eingeführt und geprüft.

EINLEITUNG

Während der Konstruktion des LISPKIT LISP-Interpreters in LISPKIT LISP [3] tauchte der Bedarf nach der Funktion RPLACA auf, ohne die die Realisierung von rekursiven LETREC-Blocke von LISPKIT LISP unmöglich wäre.

Wegen des Seiteneffekts, der in diesem Fall notwendig ist, kann RPLACA-Funktion als Benutzerfunktion nicht eingeführt werden, sondern sie muss in das LISPKIT LISP-System als eine primitive Funktion eingebaut werden.

Das so gewonnene Programmiersystem wird LISPKIT LISP-System Version ARLU genannt. Die angegebene Programmiersprache bleibt eine funktionale Programmiersprache, aber nicht mehr reinfunktional.

1. RPLACA UND RPLACD BEFEHLE

Wir erweitern die Anzahl der Befehle der SECD-Maschine [1] um die Befehle RePLAcAr und RePLAcCDr.

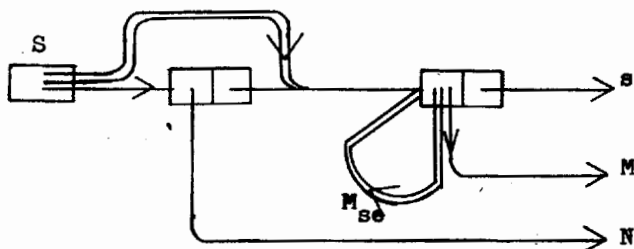
Der Befehl RPLACA kann durch den Inhalt des Wortspeichers der SECD-Maschine nicht vollständig ausgedrückt werden, wie er früher durch andere Befehle der SECD-Maschine realisiert werden konnte. Es gibt drei Fälle:

1. $(N M .s) e (RPLACA .c) d \rightarrow$ nicht definiert
wenn M ein Atom ist.
2. $(N M .s) e (RPLACA .c) d \rightarrow ((N.Y).s) e c d$
wenn M eine Liste in Form $(X.Y)$ ist
und N ein Atom oder eine Liste, für die $N \not\subseteq M$ gilt, ist.
3. $((N M .s) e (RPLACA .c) d \rightarrow (M_{se}.s) e c d$
wenn N und M die Listen sind und es gilt: $N = M$
und $N \subset M$.

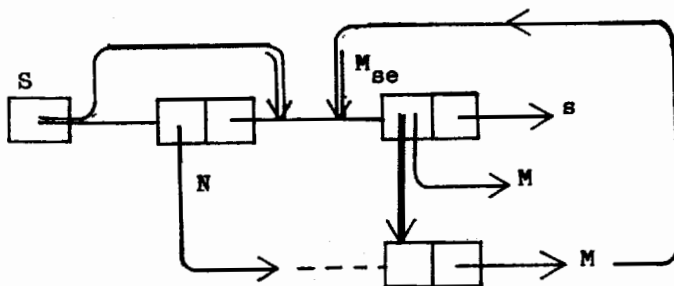
Hier ist $M_{se} \neq M$ beziehungsweise $M_{se} = M$ aber mit Seiteneffekten.

Der entstehende Seiteneffekt kann am besten graphisch dargestellt werden. Siehe die folgenden Bilder. Die Doppellinie bezeichnet den Stand nach dem ausgeführten Befehl.

N = M



N C M



Eine gleichartige Definition gilt für den Befehl RPLACD. Das folgende FORTRAN-Programmstück entspricht dem Befehl RPLACA:

```

INTEGER*2 SP, MAXSP, GLAVNON
INTEGER*2 STACK (100),ST(100)
LOGICAL*1 AR
.
.
.
310 SP = 0
M = CAR(CDR(S))
N = CAR(S)
IF(.NOT.ISATOM(M)) GOTO 3102
TYPE 311
311 FORMAT(' FUNKTION REPLACECAR IST NICHT DEFINIERT')
STOP

```

```
3102 IF(.NOT.ISATOM(N)) GOTO 3103
      MN = N
      GOTO 313
3103 IF(.NOT.ISEQUAL(M,N)) 3104
      MN = M
313   CAR(M) = MN
      S = CDR(S)
      C = CDR(C)
C     DER FOLGENDE BEFEHL MACHT EIN SPRUNG AN DEN ANFANG
C     DES UNTERPROGRAMMS EXECUTE
      GOTO 1
3104 IF(SP.LT.MAXSP) GOTO 3107
      TYPE 31
31    FORMAT('FEHLER-REPLACECAR; STACK IST SEHR KLEIN')
      STOP
3105 IF(SP.EQ.0) GOTO 3110
      AR = .FALSE.
      GLAVNON = STACK(SP)
      N = ST(SP)
      SP = SP-1
      GOTO 3109
3106 IF(AR) GOTO 3111
      CDR(GLAVNON) = M
      GOTO 3105
3107 AR = .TRUE.
      GLAVNON = N
      SP = SP+1
      STACK(SP) = GLAVNON
      ST(SP) = CDR(GLAVNON)
      N = CAR(GLAVNON)
3109 IF(ISATOM(N)) GOTO 3105
      IF(ISEQUAL(M,N)) GOTO 3106
      GOTO 3104
3110 MN = CAR(S)
      GOTO 313
3111 CAR(GLAVNON) = M
      GOTO 3105
```

Ein ähnliches Programm kann für den Befehl RPLACD auch geschrieben werden. Diese Programmbausteine wurden an den entsprechenden Stellen im Unterprogramm

EXECUTE(FN.ARGS,RESULT)

eingebaut, das den Programmsimulator der SECD-Maschine darstellt.

2. S-AUSDRUCK

2.1. Der gewöhnliche und wohlgeformte gewöhnliche S-Ausdruck

Der gewöhnliche S-Ausdruck [2] ist ein symbolischer Ausdruck. Die externe Darstellung des gewöhnlichen S-Ausdrucks wird durch folgende KF(Kontext Freie) Grammatik spezifiziert:

```

<S-Ausdruck>: : = <Atom>|
                <Molekül>
                <Molekül>: : = (<S-Ausdruck Liste>)
<S-Ausdruck Liste>: : = <S-Ausdruck>|
                        <S-Ausdruck>.<S-Ausdruck>|
                        <S-Ausdruck><S-Ausdruck Liste>

```

Die interne Darstellung des gewöhnlichen S-Ausdrucks kann graphisch als ein Baum gezeigt werden, dessen:

- innerliche Punkte Cons-Elemente
- Randpunkte Atom-Elemente sind.

Das Kennzeichen des gewöhnlichen S-Ausdrucks, wie auch jedes seiner Elemente, ist von beliebiger aber endlicher Länge. Der wohlgeformte gewöhnliche S-Ausdruck ist ein gewöhnlicher S-Ausdruck, dessen Wert in endlich vielen Schritten berechnet werden kann [3]. Der Wert eines wohlgeformten gewöhnlichen S-Ausdrucks ist ein gewöhnlicher S-Ausdruck.

2.2. Der allgemeine und wohlgeformt-allgemeine S-Ausdruck

Der allgemeine S-Ausdruck ist ein symbolischer Ausdruck, für den im Allgemeinen:

- die externe Darstellung nicht durch die angeführte kontextfrei-Grammatik spezifiziert werden kann

- die interne Darstellung graphisch nicht als ein Baum angegeben werden kann.

Eine Grammatik zur Spezifikation der externen Darstellung von allgemeinen S-Ausdrücken ist den Autoren nicht bekannt.

Die interne Darstellung allgemeiner S-Ausdrücke kann graphisch in der graphorientierten Form dargestellt werden.

Der wohlgeformte-allgemeine S-Ausdruck ist ein allgemeiner S-Ausdruck, dessen Wert ein allgemeiner S-Ausdruck ist.

2.3. Pseudo-S-Ausdruck und wohlgeformter Pseudo-S-Ausdruck

Ein Pseudo-S-Ausdruck ist ein allgemeiner S-Ausdruck, der K' ein gewöhnlicher S-Ausdruck ist.

Ein wohlgeformter Pseudo-S-Ausdruck ist ein Pseudo-S-Ausdruck, dessen Wert ein Pseudo-S-Ausdruck ist.

3. DIE GEWÖHNLICHE, ALLGEMEINE UND PSEUDO-FUNKTION

Die Funktionsargumente sind wohlgeformte, allgemeine S-Ausdrücke.

Die gewöhnliche Funktion ist eine solche Funktion, deren Wert ein gewöhnlicher S-Ausdruck ist.

Die allgemeine Funktion ist eine solche Funktion, deren Wert der allgemeine S-Ausdruck ist.

Die Pseudo-Funktion ist eine solche Funktion, deren Wert ein Pseudo-S-Ausdruck ist.

4. RPLACA UND RPLACD FUNKTIONEN

Wir erweitern die Anzahl der Funktionen um die allgemeinen Funktionen

$$(RPLACA e_1 e_2)$$

$$(RPLACD e_1 e_2)$$

Der Wert des Argumentes e_1 ist ein allgemeines Molekül, sonst sind die Funktionen undefiniert.

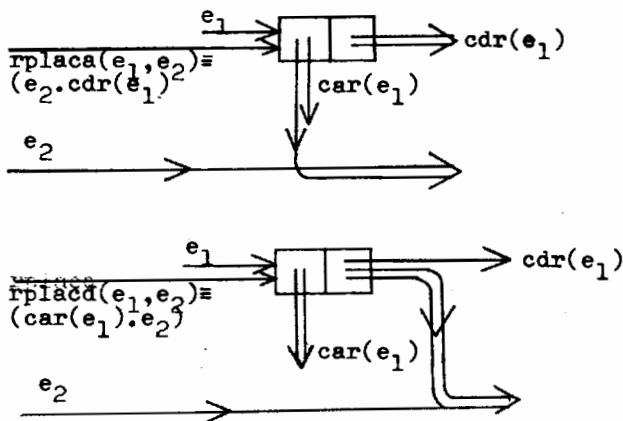
Der Wert der Funktion RPLACA ist dem Wert des allgemeinen Moleküls e_1 gleich, dessen car-Teil durch den Wert des wohlgeformten allgemeinen S-Ausdrucks e_2 ersetzt wird.

Der Wert der Funktion RPLACD ist dem Wert des allgemeinen Moleküls e_1 gleich, dessen cdr-Teil durch den Wert des wohlgeformten allgemeinen S-Ausdrucks e_2 ersetzt wird.

Bei der Berechnung des Wertes der

RPLACA-Funktion beziehungsweise der
RPLACD-Funktion

wird der Wert des ersten Argumentes e_1 der Funktion geändert, beziehungsweise, es entsteht ein Seiteneffekt, der sich aus der folgenden graphischen Darstellung ausnehmen lässt:



THEOREM

Im allgemeinen gilt es:

a) Die Werte der Ausdrücke:

$$(RPLACA \ e_1 \ e_2)$$

$$(CONS \ e_2 \ (CDR \ e_1))$$

sind untereinander verschieden,

b) Die Werte der Ausdrücke

$$(\text{RPLACD } e_1 \ e_2)$$

$$(\text{CONS } (\text{CAR } e_1) \ e_2)$$

sind untereinander verschieden.

Die Effekte der Berechnung der angeführten Ausdrücke sind untereinander verschieden.

Beweis:

Bei der Berechnung des Wertes der CONS-Funktion wird ein neues CONS-Element gebildet, weil die Inhalte der Elemente, die früher selektiert worden sind, nicht geändert werden.

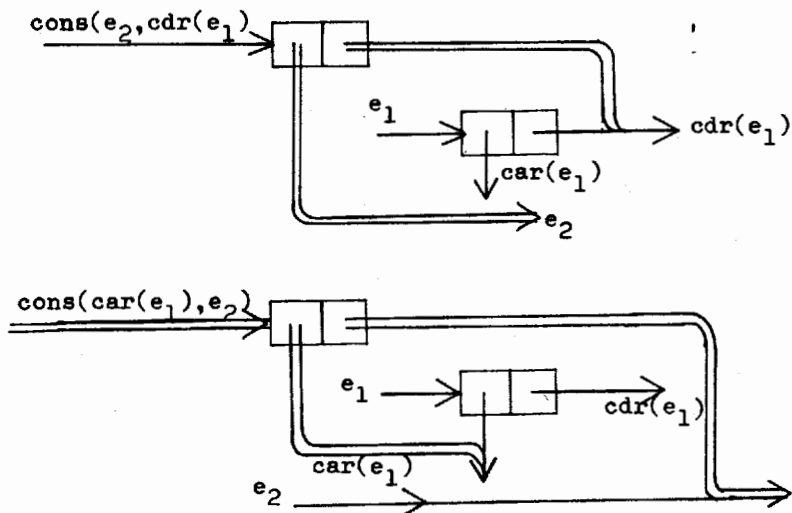
Das bedeutet, dass bei der Berechnung der Werte der Ausdrücke:

$$(\text{CONS } e_2 \ (\text{CDR } e_1))$$

und $(\text{CONS}(\text{CAR } e_1) \ e_2)$

die Werte der Ausdrücke e_1 und e_2 nicht geändert werden, also es erscheint kein Seiteneffekt.

Dasselbe ergibt sich aus der folgenden graphischen Darstellung:



Wenn

$$e_1 \neq e_2, \text{ so ist}$$

der Wert des Ausdrucks

$$(\text{RPLACA } e_1 \ e_2)$$

ein gewöhnlicher S-Ausdruck, und der Wert des Ausdrucks,

$$(\text{CONS } e_2 (\text{CDR } e_1))$$

ein gewöhnlicher S-Ausdruck, der dem vorher gebildeten Ausdruck gleich ist.

Die Effekte der Berechnung sind aber verschieden, weil bei der Berechnung des Wertes des ersten Ausdrucks der Seiteneffekt erscheint, der bei der weiteren Berechnung von Bedeutung sein kann.

Wenn

$$e_1 = e_2 = e, \text{ so ist}$$

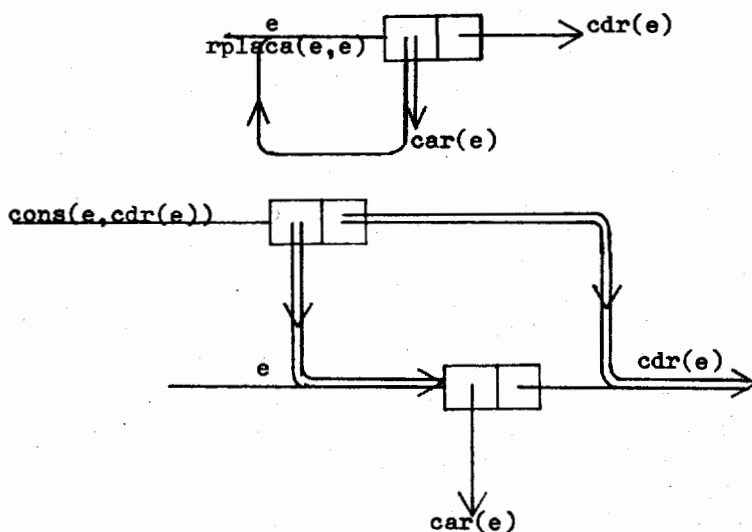
der Wert des Ausdrucks

$$(\text{RPLACA } e \ e)$$

ein Pseudo-S-Ausdruck und der Wert des Ausdrucks

$$(\text{CONS } e \ (\text{CDR } e))$$

ein gewöhnlicher S-Ausdruck. Sie sind der Definition nach verschieden. Siehe Bild 3 und 4.



Ändlich ist es, wenn

$$e_1 \neq e_2, \text{ so ist}$$

der Wert des Ausdrucks:

$$(\text{RPLACD } e_1 \ e_2)$$

ein gewöhnlicher S-Ausdruck, und der Wert des Ausdrucks

$$(\text{CONS } (\text{CAR } e_1) \ e_2)$$

ein gewöhnlicher S-Ausdruck, der dem vorher gebildeten Ausdruck gleich ist.

Die Effekte der Berechnung sind aber verschieden, weil bei der Berechnung des Wertes des ersten Ausdrucks der Seiteneffekt erscheint, der bei der weiteren Berechnung von Bedeutung sein kann.

Wenn

$$e_1 = e_2 = e, \text{ so ist}$$

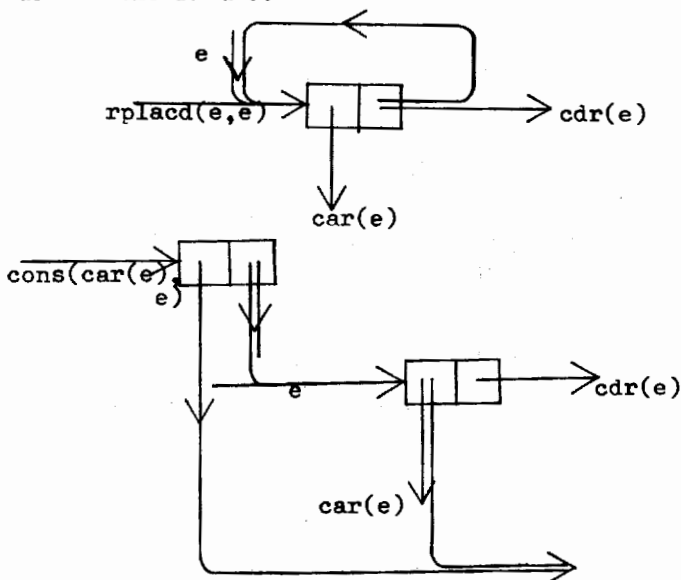
der Wert des Ausdrucks

$$(\text{RPLACD } e \ e)$$

ein Pseudo-S-Ausdruck und der Wert des Ausdrucks

$$(\text{CONS } (\text{CAR } e) \ e)$$

ein gewöhnlicher S-Ausdruck. Sie sind der Definition nach verschieden. Siehe Bild 5.



5. ÜBERSETZUNG DER FUNKTIONEN RPLACA UND RPLACD IN DIE MASCHINENORIENTIERTE SPRACHE DER SECD-MASCHINE

Sei

- e der wohlgeformte S-Ausdruck und
- n die Namenliste.

e n

ist das Ergebnis der Übersetzung des wohlgeformten Ausdrucks e in Bezug auf die Namenliste n aus LISPKIT LTSP in die maschinenorientierte Sprache der SECD-Maschine [4].

Es gelten folgende Regeln der Übersetzung der allgemeinen Funktionen RPLACA und RPLACD:

$$(RPLACA \ e_1 \ e_2) \ n = e_2 \ n \ e_1 \ n \ (RPLACA)$$

$$(RPLACD \ e_1 \ e_2) \ n = e_2 \ n \ e_1 \ n \ (RPLACD)$$

Das bedeutet:

- 1) durch die Übersetzung des Ausdrucks e_2 gewinnt man das Zielprogramm C_2 .
- 2) durch die Übersetzung des Ausdrucks e_1 gewinnt man das Zielprogramm C_1 .
- 3) durch die Konkatenation der Zielprogramme C_2, C_1 mit dem entsprechenden Befehlswort:

$$C_2 \ C_1 \ (RPLACA)$$

beziehungsweise

$$C_2 \ C_1 \ (RPLACD)$$

gewinnt man ein vollkommenes Zielprogramm.

6. MODIFIKATION DES COMPILERS DER LISPKIT LISP IN DIE MASCHINENORIENTIERTE SPRACHE DER SECD-MASCHINE

Im Einklang mit den vorher definierten Regeln der Übersetzung der allgemeinen Funktionen RPLACA und RPLACD muss man die folgende Modifikation des Compilers der LISPKIT LISP durchführen:

```
(IF (EQ (CAR E) (QUOTE RPLACD))
  (COMP (CAR (CDR (CDR E)))
    N
    (COMP (CAR (CDR E))
      N
      (CONS (QUOTE 30) C)
    )
  )
)
```

beziehungsweise

```
(IF (EQ (CAR E) (QUOTE RPLACD))
  (COMP (CAR (CDR (CDR E)))
    N
    (COMP (CAR (CDR E))
      N
      (CONS (QUOTE 31) C)
    )
  )
)
```

7. DAS VERFAHREN FÜR MODIFIKATION DES LISPKIT LISP-SYSTEMS

Die Modifikation des LISPKIT LISP-Systems muss man in folgender Reihenfolge ausführen:

1. neue Programmbausteine in das Unterprogramm EXECUTE eintragen
2. Übersetzung des LISPKIT LISP-Systems durch FORTRAN-Compilers ausführen
3. Verbindung des LISPKIT LISP-Systems durch TASK BILDER ausführen
4. Modifikation des Compilers der LISPKIT LISP
5. Übersetzung des Compilers der LISPKIT LISP.

Nach diesen Schritten ist das LISPKIT LISP-System um die allgemeinen Funktionen

```
(RPLACA e1 e2)    und
(RPLACD e1 e2)
```

erweitert und für die Arbeit bereit.

Lassen wir es durch folgendes Beispiel prüfen:

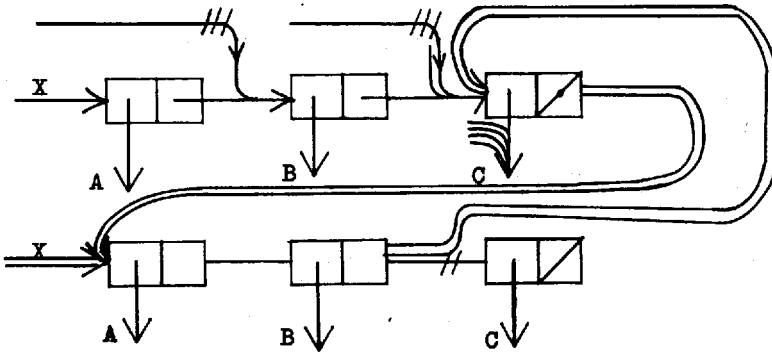
Beispiel

LISPKIT LISP-Programm P1:

```
(LET P1
  (P1. (LAMBDA (X)
    (CAR (RPLACD (CDR (CDR X)) X))))))
```

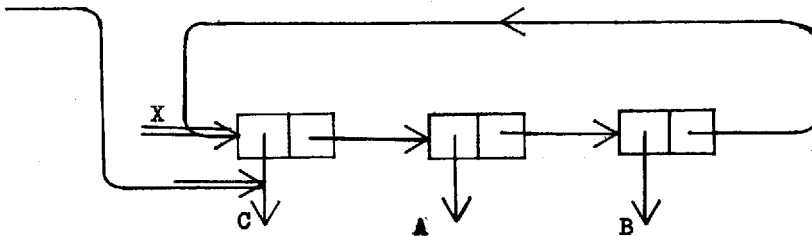
berechnet für die Eingabe $X = (A B C)$ die Ausgabe C.

Der Ablauf des Programms P1 kann durch folgende graphische Darstellung verdeutlicht werden.



Die Ordnungszahl der Programmausführung wurde durch die Linienzahl des Zeigers bezeichnet.

Der Stand nach dem Ende von Abarbeitung kann durch folgendes Bild dargestellt werden.

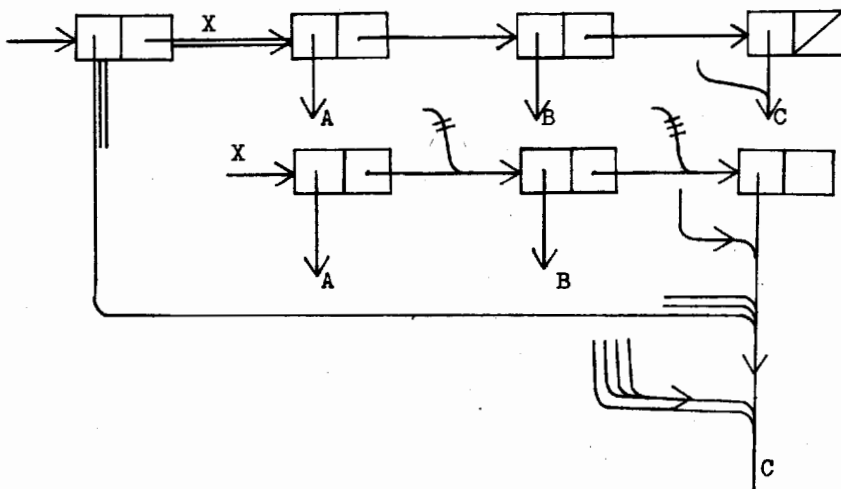


LISPKIT LISP-Programm P2:

```
(LET P2
  (P2.(LAMBDA (X)
    (CAR (CONS (CAR (CDR (CDR X))) X))))))
```

berechnet für dieselbe Eingabe $X = (A B C)$ die Ausgabe C .

Die Arbeitung des Programms P2 kann durch die folgende graphische Darstellung verdeutlicht werden.



Hierbei entstehen die folgenden Zuweisungen:

```
car(X) + C
cadr(X) + A
caddr(X) + B
cad3.dr(X) + C
cad4.dr(X) + A
cad5.dr(X) + B
.
.
.
cadn.dr(X) + cadn r(X)
```

Daraus folgt im allgemeinen

$$\text{cad}^{3k}_r(X) \rightarrow C \quad k = 0, 1, 2, \dots$$

$$\text{cad}^{3k+1}_r(X) \rightarrow A$$

$$\text{cad}^{3k+2}_r(X) \rightarrow B$$

Mit Hilfe der allgemeinen Funktionen

$$(\text{RPLACA } e_1 \ e_2)$$

und $(\text{RPLACD } e_1 \ e_2)$

lassen sich solche Transformationen der allgemeinen S-Ausdrücke ausführen die früher nicht möglich waren.

Die Funktionen

$$(\text{RPLACA } e_1 \ e_2)$$

und $(\text{RPLACD } e_1 \ e_2)$

muss man äusserst vorsichtig benutzen.

REFERENCES

- [1] V. Stojković, I. Stojmenović, J. Mirčevski, Lj. Jerinić, M. Kulaš, *Programska implementacija simulatora SECD-mašine, XXVII Jugoslovenska konferencija ETAN-a, Struga 1983.*
- [2] V. Stojković, Lj. Jerinić, I. Stojmenović, J. Mirčevski, M. Kulaš, *Implementacija NP-tehnike obrade S-izraza u FORTRAN-jeziku, 5. Medjunarodni simpozij "Kompjuter na sveučilištu", Cavtat 1983.*
- [3] V. Stojković, J. Mirčevski, *Implementacija LISP-interpretatora na LISPKIT LISP-jeziku, VIII Bosansko-hercegovački simpozijum iz informatike, "JAHORINA-84".*
- [4] I. Stojmenović, V. Stojković, Lj. Jerinić, J. Mirčevski, *O implementaciji prevodioca LISPKIT LISP-jezika na jezik SECD-mašine izvršenoj na FORTRAN-jeziku, Informatica 1, 1984.*

Received by the editors December 16, 1983.

REZIME

PROŠIRENJE LISPKIT LISP-SISTEMA SA
OPŠTIM FUNKCIJAMA RPLACA I RPLACD

U radu su definisane naredbe RPLACA i RPLACD SECD-mašine. Navedene naredbe su programski realizovane u vidu programskih segmenata napisanih na FORTRAN-jeziku i njima je proširen simulator SECD-mašine.

Definisani su obični, opšti, pseudo i njima odgovarajući valjani S-izrazi.

Definisana je obična, opšta i pseudo funkcija.

LISPKIT LISP-jezik je proširen opštim funkcijama RPLACA i RPLACD i dokazano je zbog čega su navedene funkcije opšte.

Definisano je prevodjenje opštih funkcija RPLACA i RPLACD na mašinski jezik SECD-mašine i u skladu sa tim izvršena je odgovarajuća modifikacija prevodioca.

Tako prošireni LISPKIT LISP-sistem je istestiran na računaru.