

6. REALIZACJA PROGRAMU

6.1. Wiadomości ogólne

Realizacja programu (napisanego w języku GIER-ALGOL) na maszynie GIER polega na:

- wydziurkowaniu programu na taśmie papierowej poprzez przepisanie go na flexowriterze oraz wydziurkowanie danych do programu,
- uruchomieniu programu na maszynie,
- wykonaniu obliczeń,
- opracowaniu otrzymanych wyników.

Omówimy kolejno powyższe etapy realizacji.

6.2. Dziurkowanie programu na flexowriterze

Taśmę papierową z wydziurkowanym na niej programem uzyskuje się przez przepisanie programu na flexowriterze. Przepisuje się tak, jak na maszynie do pisania, z tym że równolegle następuje perforacja kodów na taśmie i druk odpowiadających im symboli na arkuszu papieru. Druk na papierze bardzo ułatwia kontrolę dziurkowania, gdyż zamiast odczytywać kody na taśmie papierowej, porównujemy program wydrukowany na flexowriterze z programem oryginalnym.

Przy dziurkowaniu należy pamiętać o następującym:

- podkreślenia znaków uzyskuje się przez przyciśnięcie najpierw klawisza podkreślającego, tzn. klawisza ze znakiem a potem klawisza ze znakiem, który ma być podkreślony; na przykład begin uzyskujemy przyciskając kolejno klawisze

 b _ e _ g _ i _ n

(należy pamiętać, że znaki i | nie powodują przesuwu karetki);

- symbole \wedge \perp \dagger \ddagger \cdot \leq \geq \dagger \equiv \Rightarrow $-$, $:=$

uzyskujemy przyciskając k o l e j n o dwa klawisze, a mianowicie odpowiednio:

\wedge	przez uderzenie klawiszy	\wedge
\perp	- , , -	- ,
\dagger	- , , -	<
\ddagger	- , , -	>

+	- , , -	-:
≤	- , , -	-<
≥	- , , -	->
≠	- , , -	=
=	- , , -	=
=>	- , , -	=>
-,	- , , -	-,
:=	- , , -	:=

- symbole END CODE, CLEAR CODE, SUM CODE, RED RIBBON, BLACK RIBBON, Å (albo Ä) uzyskujemy przyciskając r ó w n o c z e ś n i e dwa klawisze, a mianowicie odpowiednio:

END CODE	przez uderzenie klawiszy	SPACE	AUX CODE
CLEAR CODE	- , , -	O	AUX CODE
SUM CODE	- , , -	a albo A	AUX CODE
RED RIBBON	- , , -	< albo >	AUX CODE
BLACK RIBBON	- , , -	b albo B	AUX CODE
Å albo Ä	- , , -	l albo V	AUX CODE

- ewentualne błędy można likwidować dobijając ręcznie w danym rzędku dziurki brakujące do kodu TAPE FEED albo ALL HOLES, tzn. do jednego z następujących dwu kodów:

0000:000
00000:000

ponieważ kody te są przy czytaniu przez maszynę ignorowane; należy tu pamiętać o zaznaczeniu ołówkiem na taśmie miejsca, w którym mają być później ręcznie doperforowane dziurki, ponieważ bez oznaczenia odnalezienie takiego miejsca jest bardzo kłopotliwe; ponadto należy pamiętać, że przez dobicie dziurek dany kod zostaje zniszczony, a więc bezpośrednio po omyłce należy wyperforować kod właściwy; przy pewnej wprawie można uniknąć ręcznego dobijania dziurek przez ręczne cofnięcie wyperforowanej taśmy o jeden rząd i przyciśnięcie klawisza TAPE FEED;

- przeniesienia do nowego wiersza mogą być na ogół dokonywane w innych miejscach niż w oryginalnym programie: wyjątek stanowią teksty zawarte między znakami << i >, w których bezwarunkowo przeniesienia do nowego wiersza (przez przyciśnięcie klawisza CAR RET) mogą być robione tylko w tych miejscach, w których jest to przewidziane w programie oryginalnym.

6.3. Perforowanie danych

W przypadku większej liczby danych, do programu perforujemy je z reguły na oddzielnej taśmie (ewentualnie kilku czy kilkunastu oddzielnych taśmach), gdyż ułatwia to operowanie nimi przy maszynie. Z drugiej strony w razie wykrycia omyłek w perforacji mamy wtedy do czynienia z wymianą krótszych taśm.

W przypadku korzystania z kilku taśm papierowych w jednym programie z reguły kończymy każdą z nich kodem END CODE. Jak już o tym była mowa, powoduje to później przy czytaniu przez maszynę zatrzymanie maszyny i automatyczny druk wyrazu

pause

co stanowi sygnał zakończenia czytania danej taśmy. Zasadniczo można by czytać jedną taśmę po drugiej bez użycia kodu END CODE, ale praktyka uczy, że otrzymujemy wtedy duże prawdopodobieństwo przekłamań ze strony czytnika. Dlatego zaleca się kończyć każdą taśmę (o ile nie jest to taśma końcowa) kodem END CODE.

Dziurkując taśmę z danymi musimy pamiętać o terminatorach i komentarzach (patrz paragraf 5.32).

6.4. Poprawianie taśmy

Gdy wykryjemy błąd na wyperforowanej taśmie, z reguły taśmę taką kopiujemy, poprawiając przy tym dostrzeżone błędy. Odbywa się to w ten sposób, że zakładamy wadliwą taśmę na czytnik flexowritera i puszczaemy równoległe perforację nowej taśmy i druk na papierze. Według druku orientujemy się, w którym momencie należy zatrzymać kopiowanie dla poprawienia taśmy. Po zatrzymaniu kopiowania, dalszą część treści, obejmującą miejsce poprawiane, wypisujemy ręcznie na klawiaturze, po czym ręcznie przesuwamy taśmę na czytniku do tego miejsca, które ma być następnie kopiowane i puszczaemy automatyczne kopiowanie dalej.

W pewnych przypadkach można uniknąć kopiowania taśmy, a mianowicie, gdy błąd można usunąć przez dobicie brakujących dziurek, gdyż czynność tę można bez większych trudności wykonać ręcznym perforatorem. Jeśli na przykład zamiast kodu litery c

ooo	:	oo
-----	---	----

pojawił się na taśmie kod litery a

oo	:	o
----	---	---

to przez ręczne dobicie dwu dziurek przekształcamy go w żądany kod litery c.

Nie zaleca się natomiast zaklejania nadmiernie wyperforowanych dziurek, ponieważ taka reperacja często powoduje przekłamanie w czasie czytania przez czytnik, a miejsce zaklejone często z powrotem się odkleja. Ponadto sama czynność zaklejania dziurek jest kłopotliwa i żmudna.

6.5. Oznaczanie taśm

Każda taśma musi mieć oznaczenie, gdzie jest jej początek, a gdzie koniec (porównaj paragraf 5.7), a ponadto musi mieć swój znak charakterystyczny, umożliwiający jej łatwe rozpoznanie. Można go na przykład zawsze wypisywać atramentem na początku taśmy. Szczególnie łatwo zapomina się o właściwym oznakowaniu taśmy w przypadku operowania kilkoma wariantami tego samego programu: po oznakowaniu jednego wariantu zapomina się oznakować drugiego, co utrudnia bardzo późniejsze jego rozpoznanie. Zapomina się także o oznakowaniu kopii taśmy po jej automatycznym skopiowaniu. Dla uniknięcia pomyłek zaleca się również natychmiastowe niszczenie taśm błędnych, po ich skopiowaniu połączonym z naprawą błędów.

6.6. Uruchamianie programu

Mając gotową taśmę z wyperforowanym na niej programem możemy przystąpić do uruchamiania programu na maszynie. Polega ono na następujących czynnościach:

- przygotowaniu maszyny,
- odczytaniu programu przez maszynę i automatycznym przetłumaczeniu go (tzw. translacji) na kod wewnętrzny,
- w przypadku wydrukowania przez maszynę sygnałów alarmujących o błędach w programie, odszukaniu tych błędów i poprawieniu na taśmie papierowej, po czym następuje ponowne wprowadzenie programu do maszyny,
- w przypadku niestwierdzenia przez maszynę błędów w programie, co zostaje zasygnalizowane drukiem wyrazu

run

wykonaniu ewentualnie krótkiego sprawdzającego liczenia dla kontroli jego poprawności; w wielu przypadkach rezygnujemy z tego kontrolnego liczenia, zwłaszcza wtedy, gdy kontrola może nastąpić dopiero po dłuższym liczeniu.

W większości ośrodków obliczeniowych osobne godziny są przeznaczone na uruchamianie programów, a osobne na właściwe liczenie. Uruchamianie programów odbywa się bowiem na maszynie zazwyczaj krótkimi, parominutowymi etapami, natomiast właściwe liczenie trwa zwykle dłużej. Wobec tego w pewnych godzinach maszyna jest udostępniona po kilka minut różnym użytkownikom, a w innych godzinach przyjmuje się zamówienia na pracę maszyny tylko na dłuższe okresy. Ze względu na duży koszt eksploatacji maszyn cyfrowych ważne jest zamawianie właściwego okresu czasu pracy maszyny.

Omówimy kolejno poszczególne wymienione etapy.

6.7. Przygotowanie maszyny

Maszyna GIER sygnalizuje za pośrednictwem druku na maszynie do pisania liczne charakterystyczne sytuacje. Na sygnały maszyny operator odpowiada, pisząc na tejże maszynie do pisania odpowiednie teksty. W ten sposób toczy się stale pewnego rodzaju rozmowa między operatorem a maszyną.

Sygnały maszyny można podzielić na informujące i alarmowe. Sygnałami informującymi będziemy nazywali sygnały, pojawiające się w czasie normalnego prawidłowego działania maszyny i prawidłowego programu. Sygnałami alarmowymi będziemy nazywali sygnały jakichkolwiek błędów. Sygnały informujące maszyna drukuje czarno, alarmowe czerwono.

Sygnały alarmowe można podzielić na sygnały błędów wykrytych w programie przez maszynę w czasie automatycznego tłumaczenia tego programu z języka GIER-ALGOL na kod wewnętrzny oraz sygnały alarmowe w czasie liczenia programu.

W niniejszej książce nie będziemy omawiać sygnałów alarmujących błędne działanie maszyny, jak również sygnałów informujących o stanie technicznym maszyny, odsyłając programującego w przypadku pojawienia się nieznanego mu sygnału do technika dyżurującego przy maszynie.

Przygotowanie maszyny do odczytania programu i automatycznego przetłumaczenia go na kod wewnętrzny maszyny polega na sprowadzeniu maszyny do tzw. sytuacji "algol", którą otrzymujemy w momencie automatycznego wydrukowania sygnału

algol

i zatrzymania się maszyny. Oznacza to, że translator w maszynie nie jest przygotowany do odczytania programu z taśmy papierowej założonej na czytnik i automatycznego przetłumaczenia tego programu na kod wewnętrzny maszyny.

Zazwyczaj przychodząc do maszyny zastajemy ją już w tej sytuacji, przygotowaną przez dyżurującego technika. Niezależnie od tego w paragrafie 6.9 omówimy sposób sprowadzania maszyny do sytuacji "algol".

Zamiast sygnału "algol" może również pojawić się

algol KA
algol KB
algol KC

co oznacza dodatkowo, że na małym pulpicie maszyny (ma ona dwa pulpity z wieloma klawiszami, jeden z tych pulpitów jest zawsze znacznie mniejszy od drugiego) są włączone klucze odpowiednio KA, KB albo KC. Każdy z tych kluczy jest na małym pulpicie oznaczony odpowiednimi literami. Każdemu z tych kluczy odpowiada na pulpicie dwa klawisze i zielona lampka. Klucz włącza się przyciskając prawy, a wyłącza przyciskając lewy klawisz. Gdy klucz jest włączony, odpowiednia zielona lampka pali się, gdy klucz jest wyłączony, lampka się nie pali.

Przy normalnej pracy translatora klucze KA, KB i KC powinny być wyłączone. W przypadku pojawienia się niepożądanego sygnału "algol KA", "algol KB" albo "algol KC" wyłączamy na małym pulpicie odpowiadający temu sygnałowi klucz i zasadniczo można by wtedy uznać, że maszyna jest już w sytuacji "algol". Dla późniejszej kontroli lepiej jest jednak po wyłączeniu kluczy KA, KB czy KC spowodować druk symbolu "algol"

przez sprowadzenie maszyny do tej sytuacji w sposób opisany w paragrafie 6.9.

Zanim ten sposób opiszemy, omówimy jeszcze pewne cechy translatora.

6.8. Różne postaci translatora. Możliwość powiększenia swobodnej części pamięci bębnowej

Jak już o tym była mowa, przy normalnym umieszczeniu translatora na bębnie zajmuje on wraz z liczonym programem zazwyczaj około 5800 miejsc, pozostawiając swobodnych około 7000. W niniejszym paragrafie zajmiemy się tą sprawą dokładniej.

Translator GIER ALGOL III BUF składa się z trzech części: tzw. systemu HELP zajmującego na bębnie 1560 miejsc, części stałej zajmującej 1720 miejsc i części używanej tylko do tłumaczenia programu a zajmującej 4240 miejsc, co daje łącznie 7520 zajętych miejsc na bębnie. Z powyższego wynika, że sam program przetłumaczony na język wewnętrzny maszyny nie może zajmować więcej niż 5280 miejsc na bębnie, a w czasie jego liczenia dla procedur to drum i from drum można wykorzystywać jedynie 4240 miejsc zajętych w czasie tłumaczenia programu przez translator, ale swobodnych po przetłumaczeniu programu. Zazwyczaj programy nie zajmują aż 5280 miejsc na bębnie i wtedy zwiększa się liczba miejsc swobodnych dla procedur to drum i from drum.

Powyższy podział pojemności bębna może okazać się niewystarczający z dwojakich powodów. Albo sam program jest tak długi, że po przetłumaczeniu na język wewnętrzny zajmuje więcej niż 5280 miejsc, albo w czasie jego liczenia za mało jest miejsc swobodnych dla procedur to drum i from drum.

Dla powiększenia swobodnej części pamięci bębnowej stosuje się m.in. tzw. s k r ó c o n y t r a n s l a t o r, tzn. translator bez systemu HELP, który kosztem powolniejszej pracy pozwala zaoszczędzić 1440 dodatkowych miejsc. W przypadku użycia tej postaci translatora program po przetłumaczeniu go na kod wewnętrzny może zajmować aż do 6720 miejsc na bębnie, a dla procedur to drum i from drum pozostaje swobodnych

10960 - p

miejsc, gdzie p jest liczbą miejsc zajęta przez przetłumaczony program. W przypadku maksymalnego programu zajmującego aż 6720 miejsc pozostaje tylko 4240 miejsc swobodnych, ale w przypadku programów małych liczba miejsc swobodnych znacznie się może powiększyć.

Liczbę miejsc zajętych przez program można otrzymać i wydrukować za pomocą króciutkiego następującego fragmentu programu włączonego do początku programu głównego:

```
c:= drumplace;
write(⟨aaaaa⟩,12799-c)
```

ponieważ przetłumaczony program zajmuje miejsca od nr 12799 w kierunku mniejszych numerów. Gdybyśmy chcieli otrzymać nie liczbę miejsc zajętych przez program, a liczbę miejsc swobodnych dla procedur to drum i from drum, należałoby powyższy fragment zastąpić przez:

```
c:= drumplace;
write(⟨dddd⟩,c-1839);
```

ponieważ miejsca od nr 0 do nr 1839 włącznie zajmuje skrócony translator.

Jeżeli sam program jest bardzo duży i po przetłumaczeniu musiałby zająć więcej niż 6720 miejsc na bębnie, można dla niego uzyskać jeszcze dodatkowe miejsca używając tzw. translatora z taśmą. Rzecz polega na tym, że translator odczytuje się stopniowo z taśmy papierowej albo z karuzeli i częściami wprowadza na bęben w miarę postępowania tłumaczenia programu. Sposób ten pozwala w czasie tłumaczenia programu zaoszczędzić dodatkowo 3120 miejsc na bębnie. Wtedy program przetłumaczony na kod wewnętrzny może zająć aż do 9840 miejsc, ale miejsc swobodnych dla procedur to drum i from drum pozostaje tylko jak przed tym

10960 - p

a więc w przypadku maksymalnego programu na 9840 miejsc tylko 1120 miejsc swobodnych. Za powiększenie liczby miejsc dla programu przez użycie translatora z taśmą płacimy znacznym spowolnieniem tempa translacji programu.

Fakt za małej pojemności bębna, czy to dla przetłumaczonego programu, czy dla procedury to drum, jest przez maszynę automatycznie sygnalizowany. Będzie o tym mowa w paragrafie poświęconym sygnałom maszyny.

Chcąc skorzystać z maszyny, musimy znać typ i postać używanego translatora. Translator z taśmą jest - ze względu na swą powolność - używany jedynie w wyjątkowych przypadkach. Translator zwykły jest najkorzystniejszy, jeśli liczy się z reguły programy małe z niezbyt dużym zapotrzebowaniem miejsc przez procedury to drum i from drum. Jeżeli jednak do liczenia idą często programy duże, wymagające wielkiej liczby miejsc dla procedur to drum i from drum, najczęściej stosuje się translator skrócony.

Translator GIER ALGOL III daje na bębnie o 80 miejsc swobodnych więcej niż omówiony wyżej translator GIER ALGOL III BUF.

6.9. Wywoływanie sytuacji "algol"

Sposób wywoływania sytuacji "algol" zależy od postaci translatora i od aktualnej sytuacji maszyny. Rozróżniamy trzy przypadki:

I. Maszyna wydrukowała jakikolwiek z następujących sygnałów:

alas	exp	param
array	gier	spill
drum alas	index	sqr
end	ln	

po czym zatrzymała się. W takiej sytuacji przyciśnięcie jakiegokolwiek klawisza na maszynie do pisania z wyjątkiem r, b, w, p powoduje sprowadzenie maszyny do sytuacji "algol", a więc i automatyczny druk sygnału

algol

II. Posługujemy się translatorem bez HELPu (tzn. skróconym). Wtedy z dowolnej sytuacji maszyny możemy ją sprowadzić do sytuacji "algol" przyciskając na małym pulpicie maszyny kolejno klawisze

RESET
HP

III. Posługujemy się pełnym translatorem (tzn. zawierającym system HELP). Wtedy z dowolnej sytuacji maszyny możemy ją sprowadzić do sytuacji "algol" przez przyciśnięcie na małym pulpicie maszyny kolejno klawiszy

RESET
HP

na co maszyna reaguje drukiem sygnału

hp-knap

a następnie przez wypisanie na maszynie do pisania tekstu

halgol
e

albo

heeee
e

z tym, że po każdym z powyższych wierszy należy zrobić powrót karetki przez przyciśnięcie klawisza CAR RET.

Jeżeli zamiast sygnału "algol" w którymkolwiek przypadku zostanie przez maszynę wydrukowany sygnał

gone

oznacza to, że przez program wykorzystujący za dużo miejsc na bębnie został uszkodzony translator. Należy wtedy zwrócić się do technika dyżurującego przy maszynie o ponowne wprowadzenie do niej translatora.

6.10. Translacja programu

Gdy mamy gotowy program na taśmie papierowej, a maszyna wydrukowała sygnał

algol

możemy uruchomić automatyczną translację programu, tzn. tłumaczenie go z języka GIER-ALGOL na kod wewnętrzny maszyny.

Jeśli po sygnale "algol" przyciśniemy na maszynie do pisania klawisz odstępu albo jakiegokolwiek znaku z wyjątkiem zastrzeżonych

p, w, t, o, l, n, i,

zostanie uruchomiony czytnik i tłumaczenie programu przez translator rozpocznie się.

Jeśli jednak po sygnale "algol" przyciśniemy klawisz "p" lub "w", następuje przez to selekcja urządzenia wyjściowego, na którym będą drukowane specjalne informacje w trakcie tłumaczenia programu: "p" oznacza perforator, "w" maszynę do pisania. Napisanie obu znaków, tzn. "pw", powoduje pojawienie się informacji na obu urządzeniach równolegle. Jeśli po znaku "p" lub "w", puścimy normalnie tłumaczenie programu (tzn. jeśli przyciśniemy klawisz odstępu lub jakiegokolwiek innego znaku niezastrzeżonego), to na wskazanym urządzeniu zostaną skopiowane wszystkie początkowe znaki wejściowej taśmy papierowej, aż do pierwszego wyrazu begin włącznie, oraz wszystkie znaki po ostatnim end, aż do pierwszego napotkanego średnika włącznie.

Można uzyskać jeszcze inne specjalne informacje w czasie tłumaczenia programu przed translator, jeśli po opisanej wyżej selekcji urządzenia wyjściowego, a przez uruchomieniem tłumaczenia przyciśniemy jeszcze dodatkowo inny klawisz maszyny. Jeśli mianowicie przyciśniemy klawisz:

l - będzie drukowany (albo perforowany) każdy, co dziesiąty wiersz programu oryginalnego, dla kontroli numerów wierszy programu,

o - tekst między PUNCH OFF i PUNCH ON zostanie włączony do programu,

t - maszyna będzie czekała na program nie z czytnika taśmy papierowej, lecz z maszyny do pisania (patrz paragraf 6.11),

n - sygnały alarmowe będą wydawane tylko na wskazane uprzednio urządzenie wyjściowe (normalnie, bez specjalnego przyciskania jakichkolwiek klawiszy, sygnały alarmowe otrzymujemy na maszynie do pisania),

i - w czasie tłumaczenia programu będą wyprowadzane (na wskazane urządzenie wyjściowe) liczne szczegółowe informacje o przebiegu tłumaczenia programu, dla zrozumienia których potrzebna jest jednak dokładniejsza znajomość maszyny albo pomoc operatora-technika; takie szczegółowe informacje są rzadko potrzebne, gdyż sygnały alarmowe otrzymujemy bez tego.

W czasie trwania tłumaczenia programu - niezależnie od powyżej opisanych interwencji operatora - pojawiają się sygnały następujące:

pause - jeśli w czasie czytania taśmy z programem zostanie napotkany kod END CODE (maszyna w takim przypadku zatrzymuje się i po przyciśnięciu jakiegokolwiek klawisza z wyjąt-

kiem "t", kontynuuje czytanie; przyciśnięcie natomiast klawisza "t" ma skutki, jak poprzednio opisano),

line nr off - jeśli w czasie czytania taśmy pojawił się kod PUNCH OFF, a poprzednio nie był przyciśnięty klawisz "o", tzn. ma nastąpić ignorowanie tekstu po PUNCH OFF ("nr" oznacza numer wiersza, w którym pojawił się PUNCH OFF, aktualna wartość tego numeru jest tu drukowana w miejsce nr),

line nr on - jeśli w czasie czytania taśmy pojawił się kod PUNCH ON, a poprzednio nie był przyciśnięty klawisz "o", tzn. tekst po ostatnim PUNCH OFF był do tego miejsca ignorowany (podobnie jak wyżej w miejsce "nr" jest tu drukowany aktualny numer wiersza, w którym pojawił się PUNCH ON),

run - gdy tłumaczenie programu dobiegło do końca i nie było sygnałów alarmujących błędy w programie, tzn. gdy poprawny program jest już w maszynie i jest gotów do liczenia.

Ponadto jeśli czytelnik maszyny natrafi na rządę taśmy, w którym na skutek błędnej perforacji znajduje się parzysta liczba dziurek (wyjątek stanowi BLANK TAPE i ALL HOLES, jak o tym była mowa w paragrafie 5.32), maszyna zatrzymuje się i oczekuje na poprawny symbol. Jest to tzw. k o n t r o l a p a r z y s t o ś c i. W powyższej sytuacji wystarcza poprawić taśmę i założyć tym samym miejscem na czytnik, by translacja biegła dalej. Ze względu na możliwe tu omyłki zazwyczaj wprowadza się poprawioną taśmę od nowa.

Jeśli program na taśmie był wytworzony przez maszynę poprzez instrukcje wyjścia i jest odpowiednio zaopatrzony w kody CLEAR CODE i SUM CODE, można do niego zastosować kontrolę sumową, jaka była opisana w paragrafie 5.24. Jeśli w trakcie czytania taśmy z takim programem suma kontrolna liczona w czasie czytania nie zgodzi się z wyperforowaną na taśmie, zostaje wydrukowany sygnał

sum fails

(co znaczy: suma szwankuje), analogicznie jak przy wprowadzaniu danych, po czym maszyna zatrzymuje się.

Oprócz powyższych sygnałów w czasie tłumaczenia programu drukowane są automatycznie sygnały alarmowe w przypadku napotkania przez maszynę typowego błędu składniowego w tłumaczonym programie. Należy pamiętać, że translator nie wykrywa wszystkich możliwych błędów w programie, ale tylko najczęściej spotykane, przewidywane przez autorów translatora. Lista takich błędów i odpowiadających im sygnałów jest podana w paragrafie następnym.

Wykrycie błędu składniowego przez translator i wydrukowanie odpowiedniego sygnału alarmowego nie zatrzymuje maszyny, która tłumaczy program dalej aż do końca, z ewentualnym drukiem sygnałów alarmujących dalsze błędy w programie.

Po zakończeniu tłumaczenia są możliwe dwie sytuacje:

I. Maszyna nie wykryła żadnych błędów w programie, tzn. uznała go za dobry i znalazła się w tzw. sytuacji "run", tzn. sytuacji powstałej przez wydrukowanie automatycznie sygnału

run

i zatrzymanie się maszyny. Oznacza to gotowość maszyny do liczenia wprowadzonego i przetłumaczonego programu.

II. W przypadku wykrycia co najmniej jednego błędu w programie maszyna nie przechodzi do sytuacji "run", ale wraca do sytuacji "algol".