

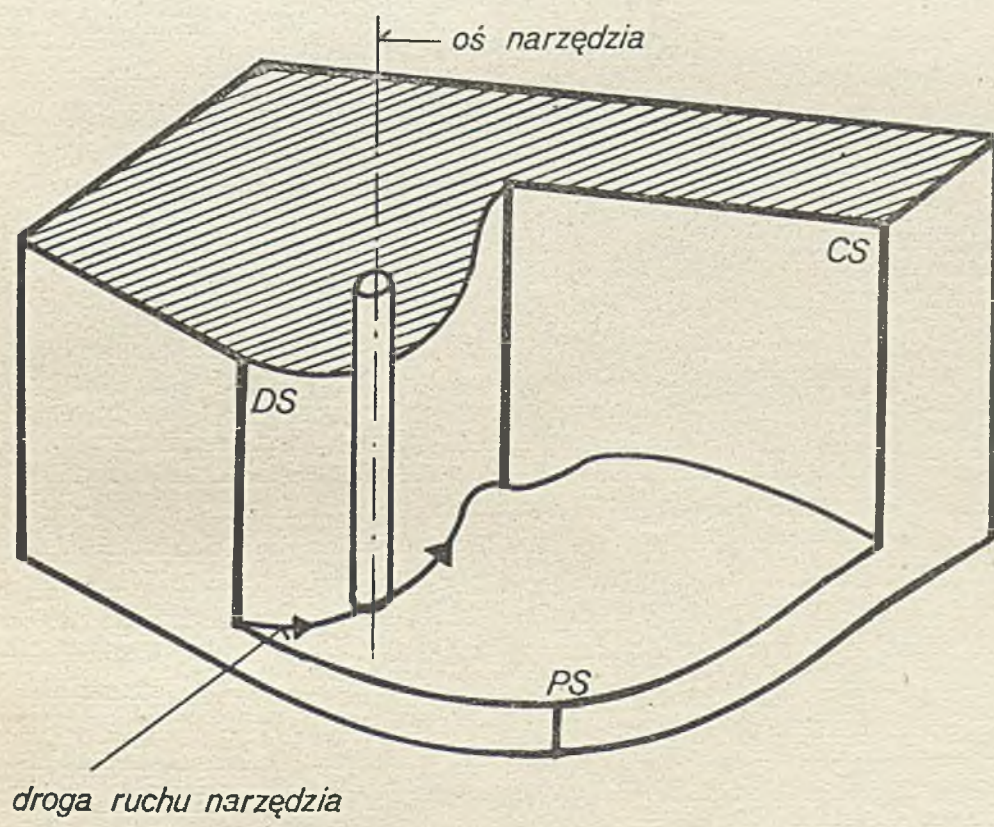
P. 3057/82

biuletyn informacyjny

5-6
'82



NAUKI
I TECHNIKI
KOMPUTEROWE



APT



P. 3057/82

Biuletyn Informacyjny NAUKI I TECHNIKI KOMPUTEROWE

Rok XX

Nr 5-6

1982

Spis treści

GUTOWSKA H: Programowanie w języku APT. Cz.III.Dodatkowe możliwości języka APT s.3

D W U M I E S I Ę C Z N I K

Wydaje:

I N S T Y T U T M A S Z Y N M A T E M A T Y C Z N Y C H

Branżowy Ośrodek Informacji Naukowej Technicznej i Ekonomicznej

Adres redakcji: ul. Krzywickiego 34, 02-078 Warszawa
tel. 28-37-29 lub 21-84-41 w. 244

Biuletyn Informacyjny NAUKI I TECHNIKI KOMPUTEROWE

mgr inż. Halina GUTOWSKA
Instytut Maszyn Matematycznych

Programowanie w języku APT

Cz. III. DODATKOWE MOŻLIWOŚCI JĘZYKA APT

Od Redakcji

Oddajemy do rąk Czytelników III - ostatnią część podręcznika
programowania obrabiarek w języku APT.

Część I - dotycząca definiowania kształtu części i detali opubli-

kowaliśmy w Biuletynie Informacyjnym OSK nr 5/6 z 1979 r.,

część II - obejmującą opisy programowania drogi narzędzia - w Biuletynie

Informacyjnym OSK nr 4 z 1981 r.

Niniejsza część III: zawiera różne dodatkowe możliwości języka APT.

Spis treści

	str.
22. Programowanie w układzie wieloosiowym	5
22.1. Instrukcja MULTAX	5
22.2. Instrukcja definiująca kierunek osi narzędzia	7
22.3. Instrukcja przekształceń specjalnych	14
23. Tworzenie pętli w języku APT	24
23.1. Instrukcja początku pętli	25
23.2. Instrukcja końca pętli	25
23.3. Instrukcja skoku warunkowego	25
23.4. Instrukcja skoku bezwarunkowego	26
23.5. Budowa pętli	26
24. Makroinstrukcje	28
24.1. Definicja początku makroinstrukcji	29
24.2. Definicja końca makroinstrukcji	29
24.3. Odwołanie się do makroinstrukcji	29
24.4. Definicje zmiennych makro	30
24.5. Budowa makroinstrukcji	34
24.6. Wielokrotne wykorzystanie etykiety	38
24.7. Makroinstrukcje systemowe	39
25. Programy specjalne w języku APT	41
26. Instrukcje wejścia /wyjścia	42
26.1. Instrukcja PRINT	42
26.2. Instrukcja PUNCH	45
26.3. Instrukcja READ	46

26.4. Instrukcje specjalne	47
27. Instrukcja definicji pomocniczej	49
28. Definicja synonimu	50
29. Struktura systemu APT	51
30. Instrukcje sterujące pracą systemu APT	53
30.1. Instrukcja CTLV	53
30.2. Instrukcja PTONLY	54
30.3. Instrukcja CLPRNT	55
30.4. Instrukcja NOPLOT	56
30.5. Instrukcja MACHIN	56
30.6. Instrukcja NOPOST	57
31. Sterowanie pracą jednostki arytmetycznej	57
31.1. Ustalanie trybu pracy jednostki arytmetycznej	57
31.2. Instrukcja ograniczająca liczbę punktów	58
31.3. Instrukcja określająca maksymalny krok	59
31.4. Instrukcje kasujące wyprowadzanie wyników	60
31.5. Instrukcja przeniesienia	63
31.6. Instrukcja obciążenia specjalnych	64
32. Instrukcje postprocesora	64
32.1. Instrukcje postprocesora bez parametrów	65
32.2. Instrukcje postprocesora z parametrami	66
32.3. Opojemna definicja szybkości posuwu	71
33. Diagnostyka w systemie APT	72
33.1. Definiowanie elementów geometrycznych w postaci kanonicznej	73
33.2. Interpretacja wyników przy definiowaniu wałka tabelarycznego	73
33.3. Instrukcja DEBUG	75
33.4. Instrukcja DYNDMP	79
33.5. Instrukcja TUNEUP	80
DODATEK A: Spis słów kluczowych języka APT	83
DODATEK B: Komunikaty o błędach fazy translacji	93
DODATEK C: Komunikaty o błędach fazy pracy jednostki arytmetycznej	99
DODATEK D: Komunikaty o błędach fazy edycji	108
DODATEK E: Suplement	110

22. PROGRAMOWANIE W UKŁADZIE WIELOOSIOWYM

W poprzednich częściach niniejszego opracowania omówione zostały możliwości języka APT pozwalające na pisanie programów obróbki części dla obrabiarek sterowanych numerycznie, w co najwyżej 3 osiach.

Język APT umożliwia też pisanie programów dla obrabiarek o bardziej złożonych układach sterowania a mianowicie sterowanych w 4 lub 5 osiach. Omówienie tego zagadnienia będzie tematem niniejszego punktu.

Obrabiarki wyposażone w wieloosiowy układ sterowania znajdują szerokie zastosowanie przy obróbce powierzchni rzeźbionych. Ustawienie wierzchołka narzędzia w obrabiarkach tego typu, w określonym położeniu związane jest z równoczesnym ruchem liniowym i obrotowym osi narzędzia podczas procesu obróbowego. Dzięki temu możliwe jest ustawienie narzędzia w dowolnym, optymalnym (z punktu widzenia procesu obróbki) położeniu, co z kolei pozwala na skrócenie drogi skrawania w porównaniu z tradycyjną obrabiarką wyposażoną w układ trójosiowy. Widać to podczas procesu obróbki powierzchni nachylonych pod kątem do płaszczyzny XY bądź też w procesie wygładzania powierzchni o nierównym kształcie. Na rys. 213 przedstawiono przykład porównania wykorzystania obrabiarki sterowanej w 5 osiach i 3 osiach do wygładzania powierzchni.

W układzie wieloosiowym możliwe jest sterowanie położeniem wierzchołka (punktu końcowego) narzędzia oraz nachyleniem osi narzędzia. Nachylenie osi narzędzia może być stałe bądź też może zmieniać się podczas przesuwania się narzędzia. Instrukcje języka APT pozwalają na takie różne sterowanie nachyleniem osi narzędzia, aby:

- oś narzędzia była nachylona pod stałym kątem w pewnym układzie współrzędnych,
- zachowywała pewien kąt z powierzchnią przedmiotu lub powierzchnią prowadzącą - może być prostopadła lub równoległa do danej płaszczyzny,
- zachowywała określony kąt z prostą prostopadłą do powierzchni przedmiotu lub do powierzchni prowadzącej oraz aby oś narzędzia zachowywała inny kąt w kierunku ruchu,
- była równoległa do prostoliniowych odcinków powierzchni prostokątnej; powierzchnia to może być zarówno powierzchnia przedmiotu, jak i powierzchnią prowadzącą.

Poniżej będą omówione instrukcje TLAXIS, VTAX oraz WCORN pozwalające na sterowanie nachyleniem osi narzędzia oraz pomocniczo instrukcja MULTAX pozwalająca na wyprowadzanie składowych wektora osi narzędzia.

22.1. Instrukcja MULTAX

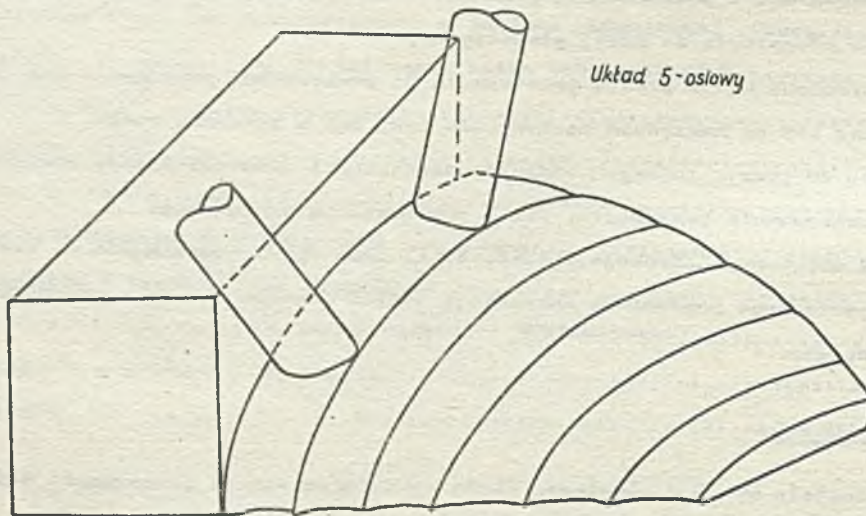
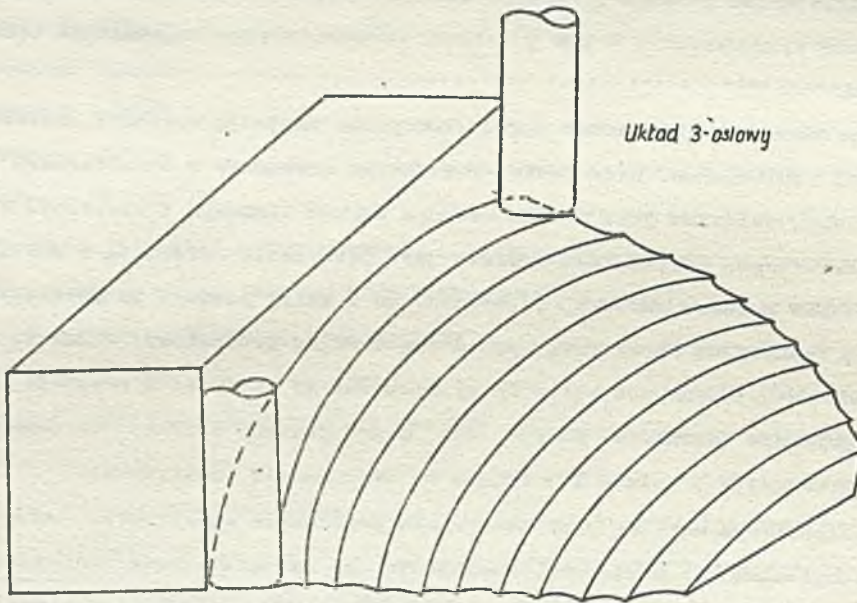
Jak to już zostało wyżej wspomniane, wiele obrabiarek oprócz przesuwania wzdłuż osi współrzędnych ma możliwość sterowania ruchem obrotowym narzędzia w jednym lub więcej kierunkach. W związku z tym system APT ma możliwość wyprowadzania, tzn. obliczania i wydrukowania nie tylko położenia punktu końcowego narzędzia (zob. pkt 11), ale także składowych osi narzędzia. Informacja ta jest później wykorzystywana przez postprocesor (zob. pkt 29) do sterowania osiami obrotowymi obrabiarki.

Instrukcja systemu APT, żądająca podania wektora osi narzędzia oraz położenia punktu końcowego narzędzia, ma następującą postać:

MULTAX

lub

MULTAX /	$\left\{ \begin{array}{c} \text{ON} \\ \text{OFF} \end{array} \right\}$



Rys. 213. Wyglądanie powierzchni w układach 3-osiowym i 5-osiowym

Pierwsza postać instrukcji wskazuje, że od momentu pojawienia się instrukcji MULTAX, w programie obróbki części informacja o wszystkich wektorach osi związanych z kolejnymi położeniami narzędzia powinny zostać przekazane do kolejnych etapów przetwarzania programu obróbki części w systemie APT.

Czasami informacja o nachyleniu osi narzędzia potrzebna jest tylko dla pewnych fragmentów programu - wówczas można skorzystać z drugiej postaci instrukcji. Wektory nachylenia osi narzędzia będą przekazywane od momentu pojawienia się w programie obróbki części instrukcji MULTAX/ON aż do momentu napotkania instrukcji MULTAX/OFF.

Jeżeli w programie obróbki części wystąpiły obie postacie instrukcji, wówczas instrukcja MULTAX ma pierwszeństwo, tzn. wektory osi narzędzia będą przekazywane dla pozostałej części programu, począwszy od chwili pojawienia się instrukcji MULTAX, ignorując np. późniejsze pojawienie się instrukcji MULTAX/ON.

Należy zwrócić uwagę, że program postprocesora, wykonywany po zakończeniu fazy obliczeń drogi narzędzia, zwykle może rozpoznawać instrukcję jedynie w pierwszej postaci, tzn. MULTAX. Oznacza to, że postprocesor oczekuje na pojawienie się wektorów osi narzędzia dla dalszej części programu, dopiero od momentu pojawienia się instrukcji MULTAX. Programista powinien upewnić się, czy postprocesor akceptuje również alternatywne postacie tej instrukcji (tj. MULTAX/ON i MULTAX/OFF).

22.2. Instrukcja definiująca kierunek osi narzędzia

Do określenia położenia osi narzędzia służy instrukcja TLAXIS. Dopuszczalnych jest kilka postaci tej instrukcji - zostaną one omówione poniżej.

W najprostszy sposób można zdefiniować nachylenie osi narzędzia, stosując instrukcję TLAXIS w następującej postaci:

TLAXIS / WE

gdzie:

WE - jest nazwą wcześniej zdefiniowanego wektora lub definicją zagnieżdżoną wektora.

Pojawienie się tej instrukcji powoduje, że oś narzędzia będzie zachowywać kierunek podany przez wektor WE dla wszystkich wektorów drogi narzędzia i wszystkich następnych instrukcji ruchu narzędzia, dopóki kierunek osi narzędzia nie będzie w inny sposób zdefiniowany (np. przez podanie następnej instrukcji TLAXIS).

Aby określić, że oś narzędzia ma być prostopadła do jednej z powierzchni charakterystycznych, należy podać instrukcję TLAXIS w następującej postaci:

TLAXIS / $\left[\begin{array}{c} \text{NORMPS} \\ \text{NORMDS} \end{array} \right]$

gdzie:

NORMPS - oznacza, że oś narzędzia ma być prostopadła do powierzchni przedmiotu,

NORMDS - oznacza, że oś narzędzia ma być prostopadła do powierzchni prowadzącej.

Gdy powierzchnie te nie są płaszczyznami, podczas ruchu narzędzia nachylenie osi narzędzia ulega zmianie tak, aby żądany warunek był stale spełniony.

Podanie instrukcji TLAXIS w postaci:

TLAXIS/1

oznacza, że ostatni kierunek osi narzędzia ma być zachowany aż do chwili podania nowego kierunku osi narzędzia.

Kierunek osi narzędzia można również określić przez nałożenie warunku, aby była ona równoległa do prostoliniowych odcinków powierzchni protokreślnej podanej w instrukcji ruchu. Instrukcja TLAXIS ma wówczas postać:

$$TLAXIS/PARLEL, \begin{Bmatrix} 1 \\ 2 \end{Bmatrix}$$

gdzie

PARLEL - jest modyfikatorem określającym, że oś narzędzia ma być równoległa do odcinków powierzchni prostokreślnej podanej w instrukcji ruchu,

1 - oznacza, że powierzchnia, względem której ma być spełniony warunek równoległości jest powierzchnią przedmiotu,

2 - wskazuje, że powierzchnią tą ma być powierzchnia prowadząca.

Najbardziej złożoną postacią instrukcji TLAXIS jest:

$$TLAXIS / ATANGL, \begin{Bmatrix} 1 \\ 2 \end{Bmatrix}, \alpha, \begin{Bmatrix} \underline{WE} \\ CUTANG \end{Bmatrix}$$

gdzie

ATANGL - jest modyfikatorem oznaczającym, że oś narzędzia zostanie ustawiona pod określonym poniżej kątem;

α - jest kątem, jaki ma tworzyć oś narzędzia z prostopadłą do powierzchni kontrolnej (por. rys. 216).

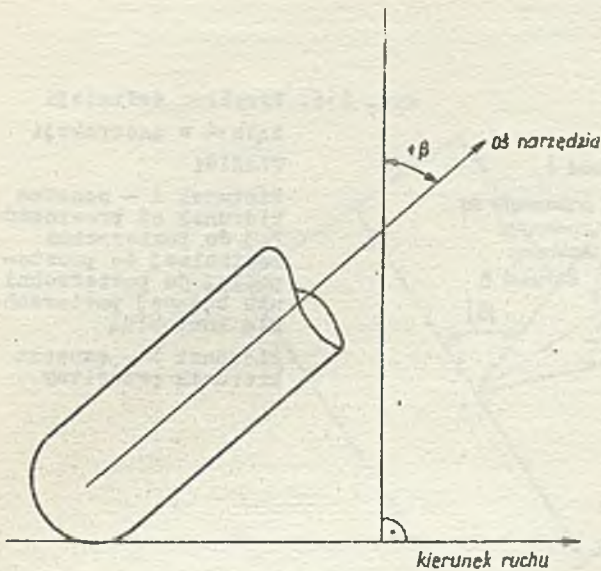
1 - oznacza, że powierzchnią kontrolną ma być powierzchnia przedmiotu,

2 - oznacza, że powierzchnią kontrolną ma być powierzchnia prowadząca,

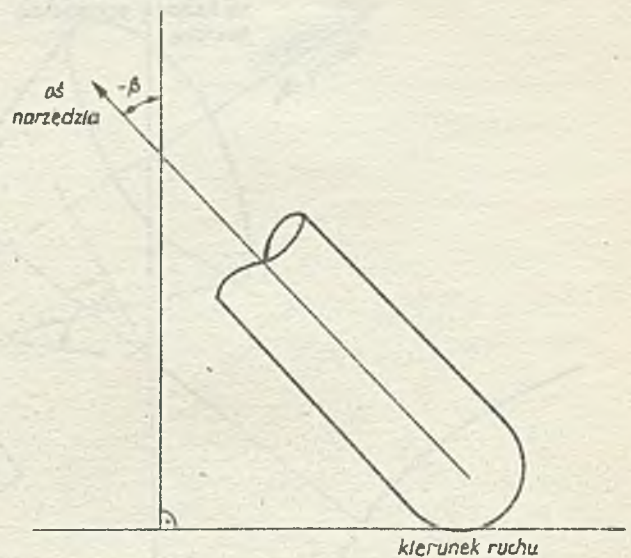
\underline{WE} - jest nazwą wcześniej zdefiniowanego wektora lub definicją zagnieżdżoną wektora; podanie wektora oznacza, że kąt nachylenia osi narzędzia może być zmieniany, ale ma ona zawsze leżeć w płaszczyźnie prostopadłej do danego wektora \underline{WE} .

CUTANG, β - oznacza, że narzędzie ma zachowywać podany kąt β z kierunkiem ruchu. Jeżeli kąt β jest dodatni, to narzędzie jest nachylone "do przodu" i tworzy tzw. kąt prowadzący z kierunkiem ruchu, natomiast gdy kąt β jest ujemny, wówczas narzędzie jest nachylone "do tyłu" względem kierunku ruchu i tworzy z nim tzw. kąt pozostający (rys. 214 i 215).

Tak więc powyższa instrukcja określa, że oś narzędzia ma tworzyć podany kąt z prostopadłą do powierzchni kontrolnej, którą może być powierzchnia przedmiotu lub powierzchnia prowadząca. Kąt α jest mierzony od prostopadłej do powierzchni kontrolnej i ma on wartość dodatnią, gdy jest skierowany w kierunku prostopadłej do drugiej powierzchni podanej w instrukcji ruchu, nie będącej powierzchnią kontrolną. Dokładniej sposób określania znaku kąta α jest przedstawiony na rys. 216.



Rys. 214. Przykład definicji dodatniego kąta β w instrukcji TLAXIS /kąt prowadzący/



Rys. 215. Przykład definicji ujemnego kąta β w instrukcji TLAXIS /kąt pozostały/

Instrukcja TLAXIS wymaga, aby był zachowany kąt α z prostopadłą do powierzchni kontrolnej a ponadto, aby osie narzędzia leżały w płaszczyźnie prostopadłej do kierunku ruchu "do przodu". Odpowiedni przykład, dokładniej wyjaśniający powyższe stwierdzenie, przedstawia rys. 217. Widać na nim, że rodzina linii tworzących kąt α z prostopadłą do powierzchni kontrolnej definiuje stożek. Przecięcie tego stożka z płaszczyzną PL, którą jest prostopadła do kierunku ruchu daje dwa wektory: V1 oraz V2. Znak kąta α określa, który z tych wektorów ma być wybrany jako wektor określający kierunek osi narzędzia - gdy kąt α jest dodatni, wówczas zostaje wybrany wektor V1, natomiast gdy jest on ujemny - wektor V2.

Pozostała część instrukcji związana jest z dodatkową kontrolą orientacji osi narzędzia. Podanie wektora WE oznacza, że kąt nachylenia osi narzędzia (zdefiniowany w pierwszej części instrukcji) może być zmieniany, ale oś narzędzia ma zawsze leżeć w płaszczyźnie prostopadłej do tego wektora. Jeżeli więc obrabiarka pracuje w układzie czteroośiowym, tzn. oprócz sterowania przesunięciem narzędzia wzdłuż trzech podstawowych osi OX, OY, OZ można również zdefiniować ruch obrotowy, to wówczas obrót będzie odbywał się w jednej płaszczyźnie.

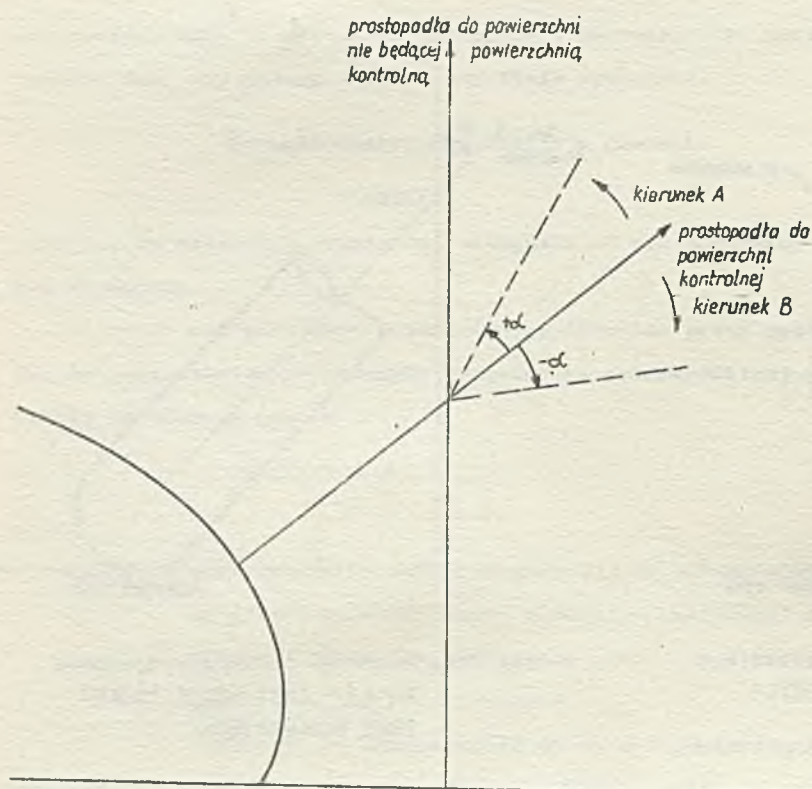
Zamiast wektora WE w instrukcji TLAXIS programista może podać kąt β , pod którym narzędzie ma być nachylone względem kierunku ruchu (rys. 214 i 215).

Należy zwrócić uwagę, że podanie samego kąta β nie definiuje jednoznacznie kierunku osi narzędzia, lecz stożek wzdłuż kierunku ruchu, w którym połowa kąta wierzchołkowego wynosi:

$$90^\circ - |\beta|$$

Przykład takiego stożka jest przedstawiony na rys. 218. Przecięcie stożka wyznaczonego przez kąt β ze stożkiem wyznaczonym przez kąt α wyznacza dwa wektory.

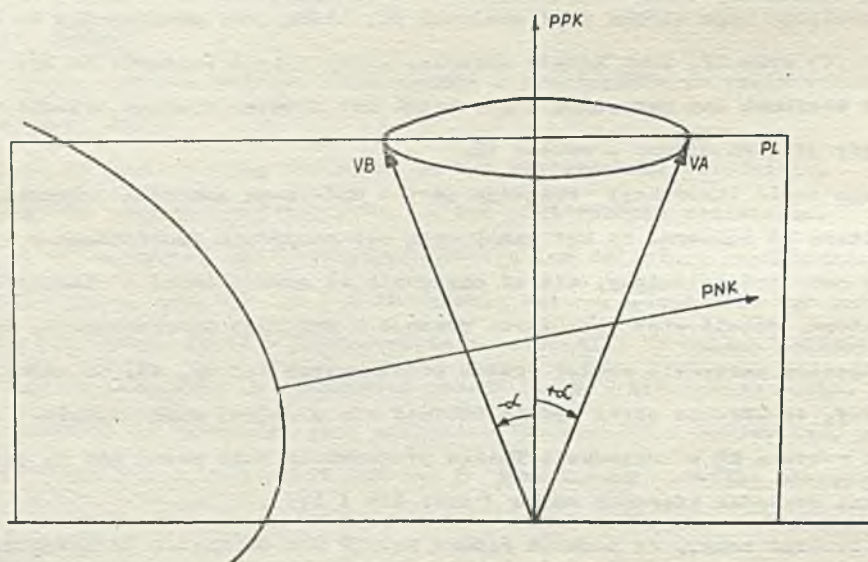
Znak kąta α pozwala na wybranie jednego z tych wektorów jako wektora jednoznacznie określającego kierunek osi narzędzia. Odpowiedni przykład jest przedstawiony na rys. 221.



Rys. 216. Przykład definicji kąta α w instrukcji TLAXIS;

kierunek A - oznacza kierunek od prostopadłej do powierzchni kontrolnej do prostopadłej do powierzchni nie będącej powierzchnią kontrolną

kierunek B - oznacza kierunek przeciwny



Rys. 217. Przykład wykorzystania znaku kąta α do określeniażądanego kierunku osi narzędzia w instrukcji TLAXIS

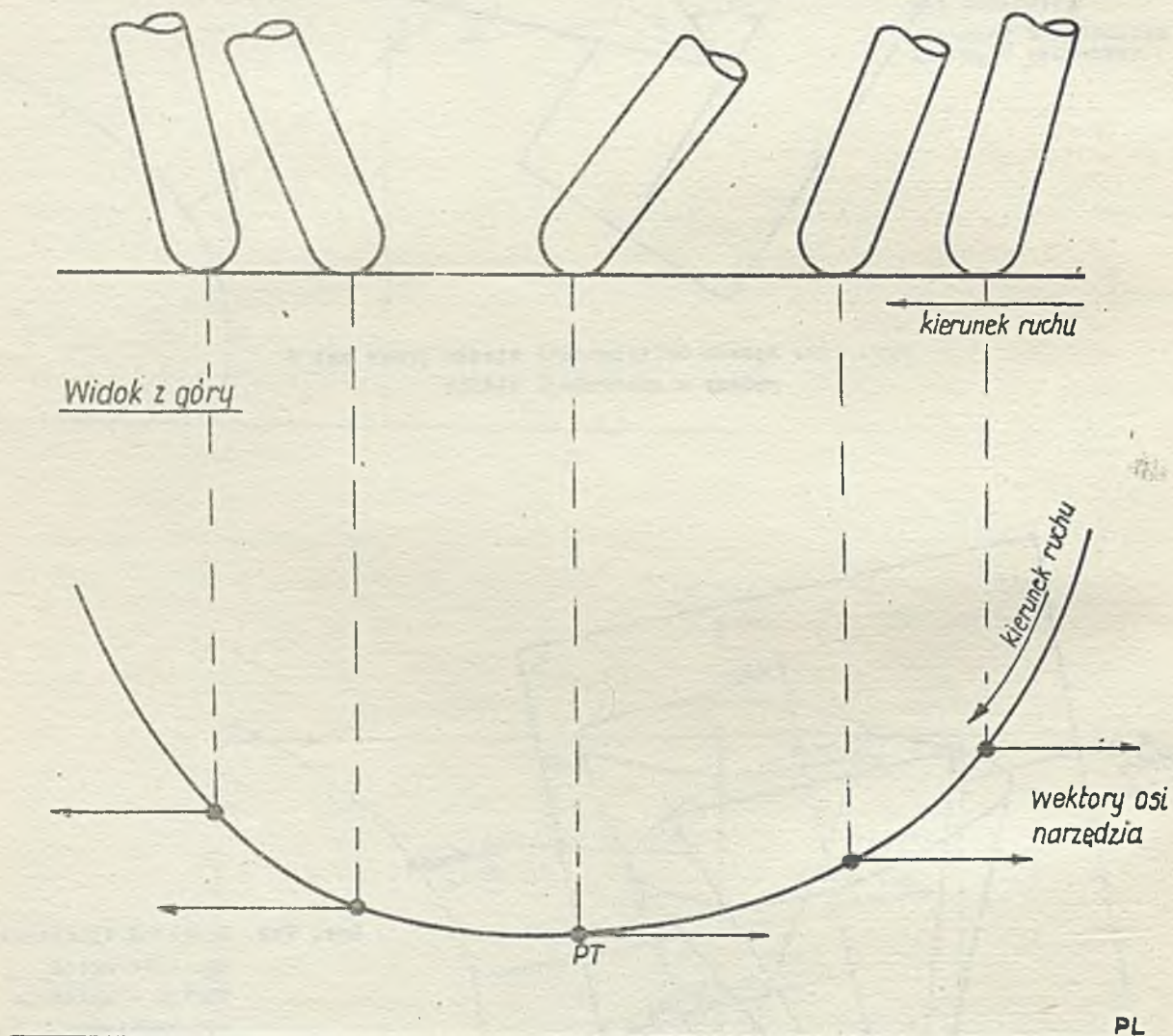
PPK - prostopadła do powierzchni kontrolnej

PNK - prostopadła do powierzchni nie będącej powierzchnią kontrolną

PL - płaszczyzna prostopadła do kierunku ruchu/leży w płaszczyźnie rysunku/

Widok z boku

(oś narzędzia ma być równoległa do płaszczyzny rysunku)



Rys. 220. Przykład sytuacji, w której kąt α jednoznacznie nie określa kierunku osi narzędzia

PL - płaszczyzna, do której ma być równoległa oś narzędzia,

PT - punkt, w którym konwencja określania kierunku na podstawie znaku kąta α staje się nieokreślona.

• Wykorzystanie instrukcji TLAXIS w układzie cztero-osiowym

Dla obrabiarki sterowanej w 4 osiach oś narzędzia ma tworzyć określony kąt (α) z prostopadłą do powierzchni kontrolnej i ma również leżeć w płaszczyźnie równoległej do pewnej płaszczyzny określonej w układzie czteroosiowym. Przykład określenia kierunku osi narzędzia w układzie czteroosiowym przedstawia rys. 219.

Poprzez podanie w instrukcji TLAXIS kąta α programista definiuje stożek, na powierzchni którego mogą leżeć osie narzędzia. Dodatkowo przez podanie wektora WE zostaje określona płaszczyzna, w której ma leżeć oś narzędzia. Ogólnie możliwe są dwa przecięcia stożka, zdefiniowanego przez podanie kąta α z płaszczyzną, zdefiniowaną przez wektor WE.

Prawidłowy kierunek osi narzędzia jest wówczas wybierany przez system APT na podstawie znaku kąta α . Ten sposób wyboru był już wcześniej omówiony. Należy zwrócić uwagę, że w jednej sytuacji nie można w powyższy sposób określić kierunku osi narzędzia: występuje ona wtedy, gdy płaszczyzna, w której ma leżeć oś narzędzia, zawiera zarówno prostopadłą do powierzchni jak i wektor określający kierunek ruchu "do przodu".

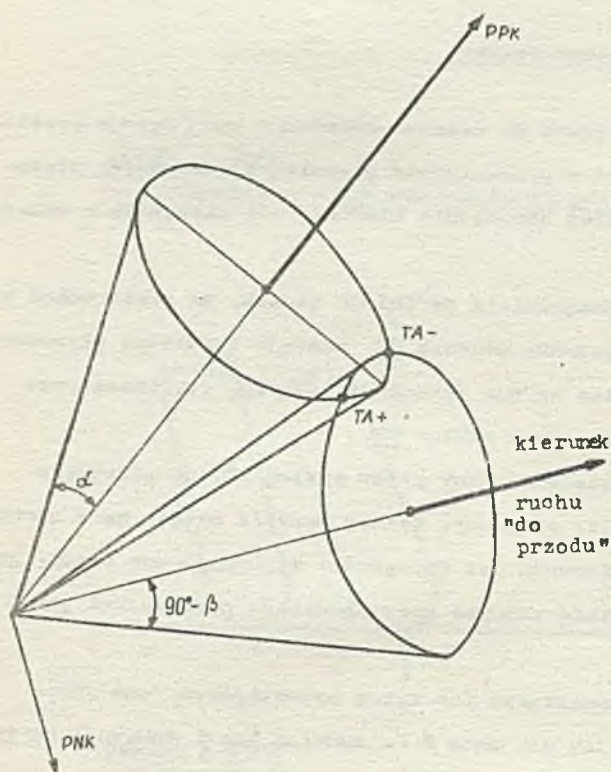
W takiej sytuacji kąt α staje się kątem prowadzącym lub kątem pozostającym, zaś wybór odpowiedniego kierunku dokonywany jest podobnie jak dla kąta β - dodatni kąt α wskazuje na kąt prowadzący, ujemny - na kąt pozostający (rys. 220).

• Wykorzystanie instrukcji TLAXIS w układzie pięcioosiowym

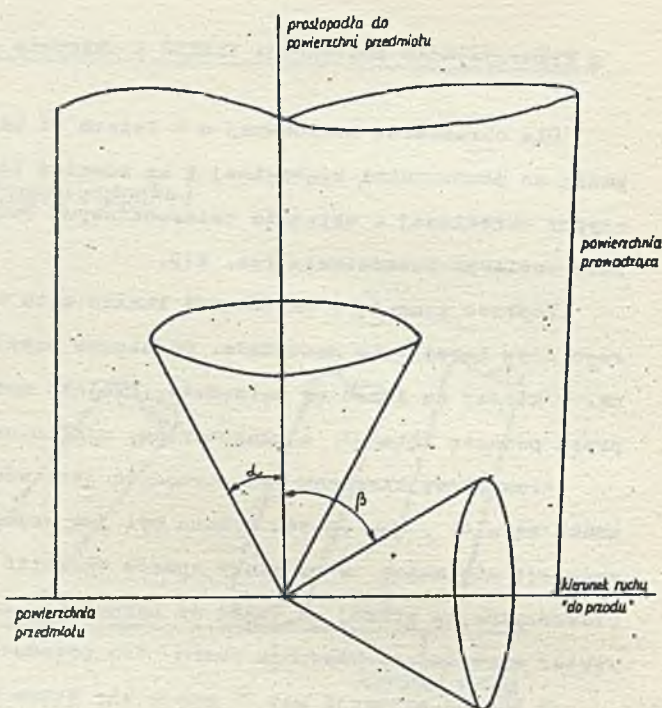
W celu zdefiniowania położenia osi narzędzia w układzie pięcioosiowym należy wykorzystać postać instrukcji TLAXIS podającą dwa kąty - α i β . Jak to już zostało wcześniej omówiono, wszystkie rozwiązania spełniające wymagania ze względu na kąt α , leżą na powierzchni stożka zdefiniowanego przez ten kąt. Podobnie wszystkie możliwe rozwiązania ze względu na kąt β leżą na stożku zdefiniowanym przez ten kąt. Na przecięciu tych stożków leżą dwa wektory, które spełniają wymagania zarówno na kąt α jak i β .

W tej sytuacji należy dokonać wyboru jednego z tych wektorów jako określającego kierunek osi narzędzia. Wybór ten jest dokonywany na podstawie znaku kąta α - w sposób opisany wcześniej przy omawianiu ogólnej postaci instrukcji TLAXIS. Odpowiedni przykład jest przedstawiony na rys. 221.

Przy stosowaniu instrukcji TLAXIS należy zwrócić uwagę na fakt, że można zdefiniować warunki, które są matematycznie niemożliwe do spełnienia. Przykład takiej niekonsekwencji definicji został przedstawiony na rys. 222. Podany w instrukcji kąt α (utworzony z prostopadłą do powierz-



Rys. 221. Przykład wykorzystania instrukcji TLAXIS w układzie pięcioosiowym
 PPK – prostopadła do powierzchni kontrolnej
 PNK – prostopadła do powierzchni, której nie jest powierzchnią kontrolną
 TA+ – położenie osi narzędzia, gdy kąt α jest dodatni
 TA- – położenie osi narzędzia, gdy kąt α jest ujemny



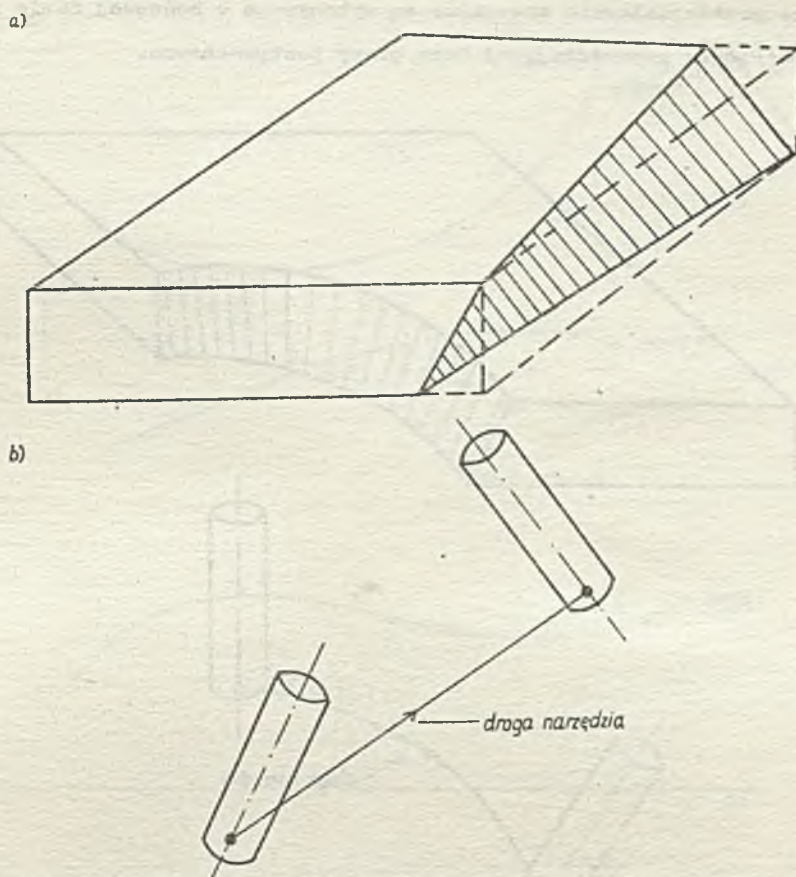
Rys. 222. Ilustracja sytuacji, gdy instrukcja TLAXIS o postaci TLAXIS/ATANG1, 1, α , CUTANG, β nie definiuje osi narzędzia

ohni przedmiotu) i kąt β (w tym wypadku kąt prowadzący) są tak określone, że zdefiniowane przez nie stożki nie przecinają się. Dlatego też nie można spełnić warunku podanego w instrukcji TLAXIS.

Ze względu na bardzo złożony sposób, w jaki system APT realizuje instrukcję TLAXIS, nie można wymienić tu wszystkich takich warunków i jako jedyną zasadę można tylko podać, aby programista uważał przy określaniu parametrów w instrukcji TLAXIS.

22.3. Instrukcje przekształceń specjalnych

Przekształcenia specjalne pozwalają na zmianę nachylenia osi narzędzia. Zmiana ta jest określana jako liniowa funkcja długości drogi pokonanej już przez narzędzie. System APT oblicza najpierw żadaną drogę narzędzia, a następnie przetwarza obliczoną już drogę w taki sposób, aby zachować początkowy kąt nachylenia osi narzędzia, końcowy kąt nachylenia osi narzędzia oraz liniową zmianę kąta nachylenia wzdłuż całej drogi.



Rys. 223. Przykład części wykonywanej za pomocą instrukcji VTLAXS
 a/ żądany kształt części
 b/ droga narzędzia podczas obróbki

Przy stosowaniu przekształceń specjalnych muszą być spełnione następujące warunki:

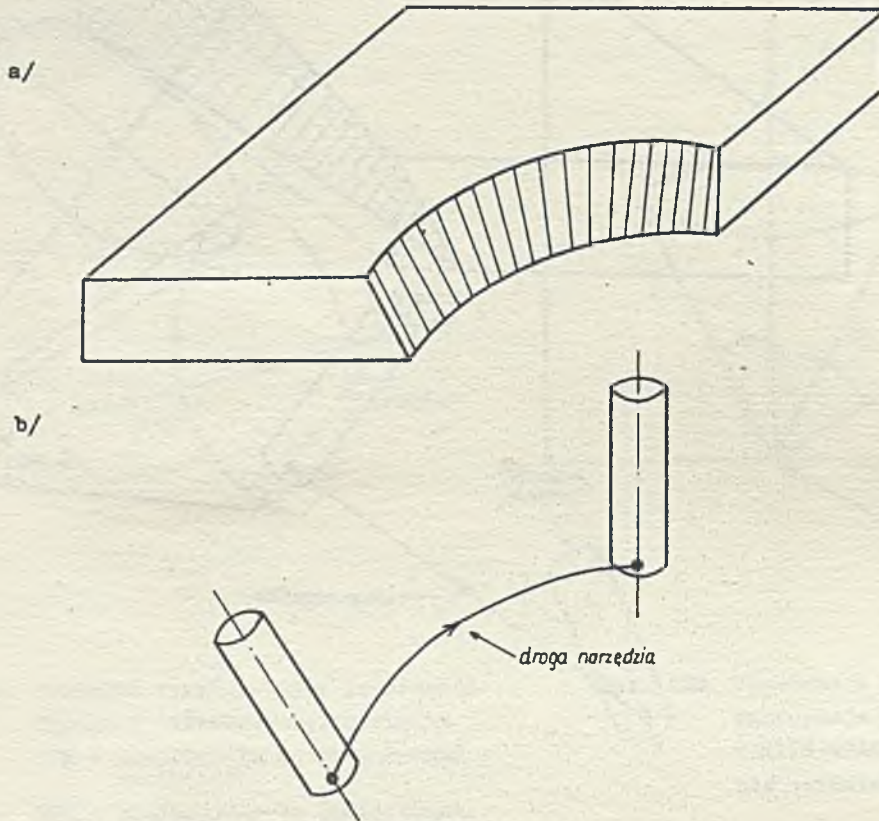
- droga narzędzia przed przekształceniem musi mieć stałą współrzędną "z",
- w programie obróbki części musi być podana instrukcja MULTAX oraz przed podaniem instrukcji przekształcenia specjalnego i wzdłuż całej drogi podlegającej transformacji, wektor osi narzędzia musi być wektorem $(0,0,1)$,
- droga narzędzia nie może zawierać żadnych punktów przegięcia.

Na rys. 225 przedstawiono przykłady dróg narzędzia (drogi A i B), które mogą być poddawane przekształceniom specjalnym oraz drogę C, która takiemu przekształceniu nie może zostać poddana, ponieważ zawiera punkt przegięcia.

W języku APT są dwie instrukcje przekształceń specjalnych - VTLAXS oraz WCORN, których funkcje będą omówione w dalszej części tego punktu.

Należy podkreślić, że instrukcje przekształceń specjalnych - podobnie jak instrukcje TRACUT i COPY (zob. pkt 19 część II podręcznika) - powodują odwołanie się do wcześniej obliczonych punktów drogi narzędzia, a nie do definicji geometrycznych lub instrukcji ruchu. Należy zdawać

sobie sprawę, że przekształcenia specjalne są wykonywane w końcowej fazie przetwarzania przez system APT, bezpośrednio poprzedzającej fazę pracy postprocesora.

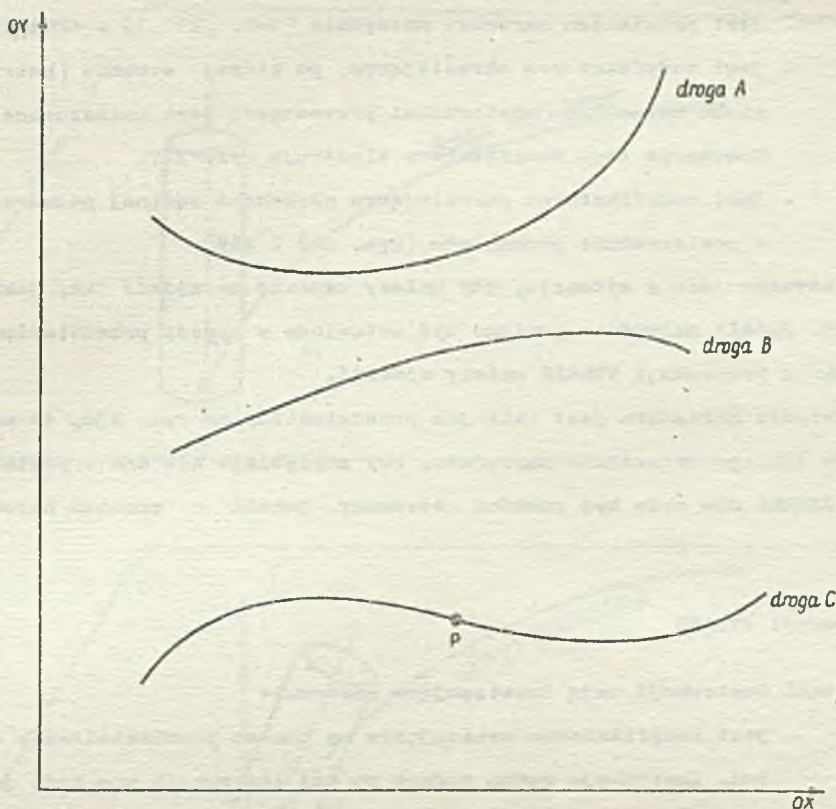


Rys. 224. Przykład części wykonywanej za pomocą instrukcji VTLAXS i WCORN
 a/ żądany kształt części
 b/ droga narzędzia podczas obróbki

● Instrukcja VTLAXS

Instrukcja VTLAXS ma następującą postać:

$$\text{VTLAXS} / \left[\begin{array}{c} \text{ON} , \left[\begin{array}{c} \text{RIGHT} \\ \text{LEFT} \end{array} \right] , \alpha , d , r , \left[\begin{array}{c} \text{RIGHT} \\ \text{LEFT} \end{array} \right] \\ \text{OFF} , \left[\begin{array}{c} \text{RIGHT} \\ \text{LEFT} \end{array} \right] , \beta \end{array} \right] \left[, \text{ZSMALL} \right]$$



Rys. 225. Przykłady dopuszczalnych i niedopuszczalnych dla przekształceń specjalnych dróg narzędzia

P - punkt przegięcia

Jak widać instrukcja VTLAXS może występować w dwóch postaciach, które będziemy umownie nazywać postacią ON i postacią OFF - od nazwy modyfikatora występującego w instrukcji bezpośrednio po ukośnej kresce. Znaczenie poszczególnych parametrów instrukcji będzie dokładnie omówione poniżej.

• Postać ON instrukcji VTLAXS

Parametry występujące w tej postaci instrukcji mają następujące znaczenie:

- ON - jest modyfikatorem wskazującym, że następne instrukcje, występujące po instrukcji VTLAXS, definiują drogę narzędzia, która to droga zostanie przekształcona za pomocą instrukcji VTLAXS.
- RIGHT lub LEFT (2 parametr) - są modyfikatorami wskazującymi początkowy kierunek (w prawo lub w lewo), w którym ma się zmienić kąt nachylenia narzędzia względem położenia pionowego - jeśli poruszamy się zgodnie z kierunkiem ruchu narzędzia. Znaczenie tego modyfikatora ilustruje rys. 226.
- α - jest początkowym kątem nachylenia osi narzędzia (podany w stopniach), liczonym od położenia pionowego, kąt ten musi być mniejszy od 80°
- d - jest średnicą narzędzia

- r - jest promieniem narożnym narzędzia (zob. pkt 10 - definicja narzędzia).
- RIGHT lub LEFT - jest modyfikatorem określającym, po której stronie (patrząc w kierunku 6-parametr
- 6-parametr - powierzchni prowadzącej jest umieszczone narzędzie. Znaczenie tego modyfikatora ilustruje rys. 227.
- ZSMALL - jest modyfikatorem określającym głębokość żądanej penetracji narzędzia w powierzchnię przedmiotu (rys. 228 i 229).

Modyfikator ZSMALL używany jest w sytuacji, gdy należy ustawić narzędzie tak, jak jest to przedstawione na rys. 228. Jeżeli narzędzie powinno być ustawione w sposób przedstawiony na rys. 229, to modyfikator ZSMALL w instrukcji VTLAXS należy opuścić.

Jeżeli kąt ustawienia narzędzia jest taki jak przedstawiony na rys. 230, to modyfikator ZSMALL nie spowoduje takiego ustawienia narzędzia, aby zagłębiało się ono w powierzchnię przedmiotu. Modyfikator ZSMALL nie może być również stosowany, jeżeli r promień narożny narzędzia jest różny od zera.

• Postać OFF instrukcji VTLAXS

Parametry w tej postaci instrukcji mają następujące znaczenie

- OFF - jest modyfikatorem wskazującym na koniec przekształcenia drogi narzędzia, tzn. instrukcje ruchu podane po tej instrukcji nie będą już przekształcane.
- RIGHT lub LEFT - są modyfikatorami określającymi, w którą stronę (w prawo lub w lewo) względem położenia pionowego ma być odchylona oś narzędzia w jego położeniu końcowym; stronę określamy patrząc w kierunku ruchu narzędzia.
- β - jest kątem nachylenia narzędzia w jego końcowym położeniu.

Aby zastosować instrukcję VTLAXS w celu przekształcenia drogi narzędzia należy podać dwie instrukcje VTLAXS, tzn. w postaci ON oraz w postaci OFF. Wówczas instrukcje ruchu zawarte między tymi dwiema instrukcjami VTLAXS definiują fragment drogi narzędzia, który będzie podlegał przekształceniu. Ilustruje to poniższy przykład.

Przykład 1

We fragmencie programu obróbki części

VTLAXS / ON , ...

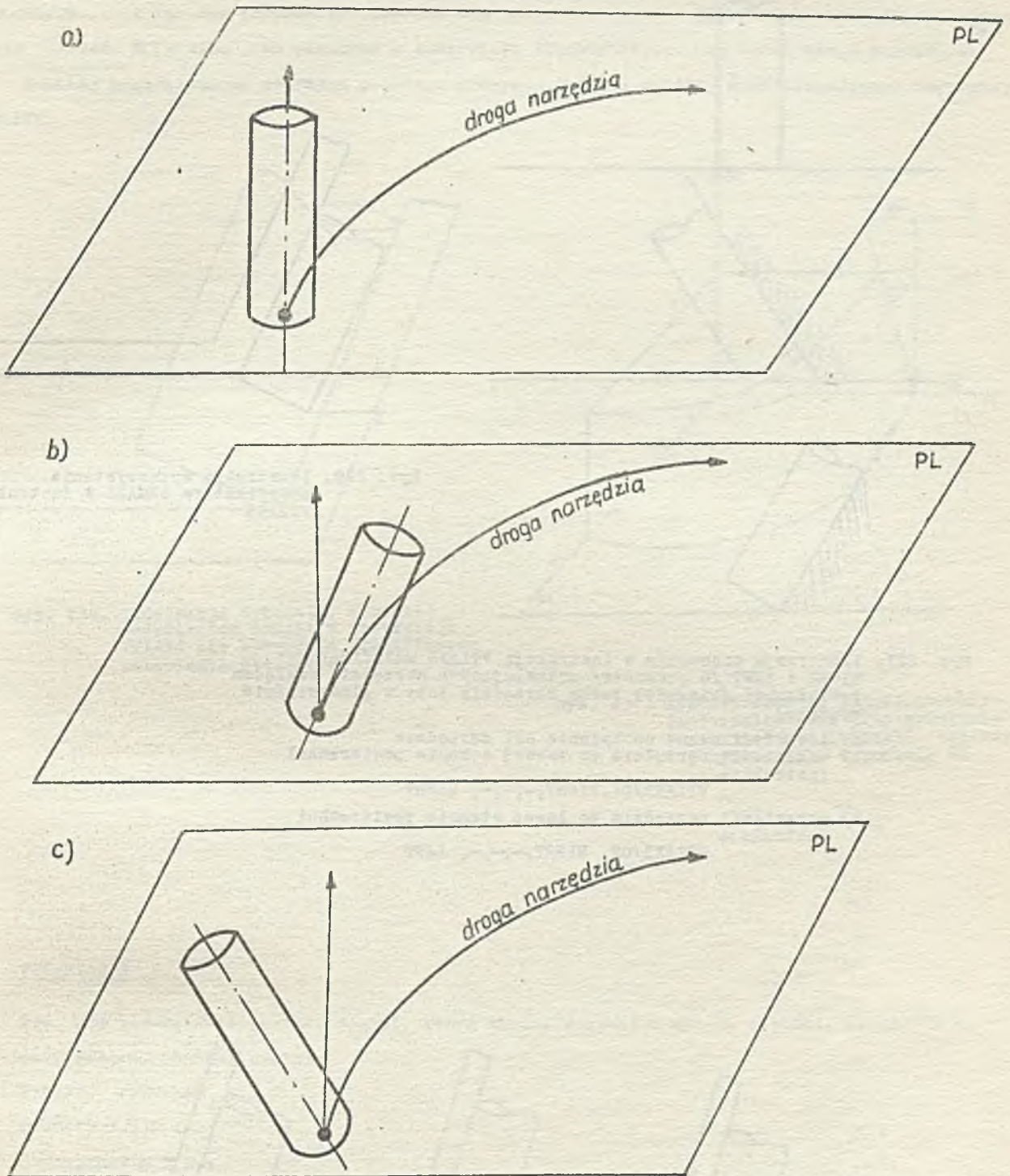
⋮

instrukcje ruchu narzędzia

VTLAXS / OFF,

droga narzędzia opisana przez instrukcje ruchu, zawarta między dwiema instrukcjami VTLAXS, będzie przekształcona w sposób opisany przez te instrukcje VTLAXS.

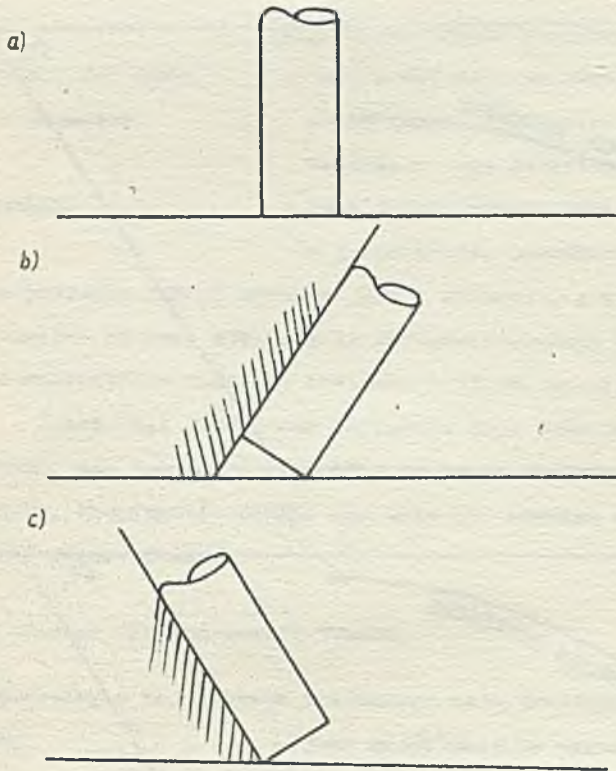
Efektom wykonania pary instrukcji VTLAXS jest więc uzyskanie drogi narzędzia, przy której początkowy kąt nachylenia osi narzędzia wynosi α (kąt ten podany jest jako parametr w instrukcji



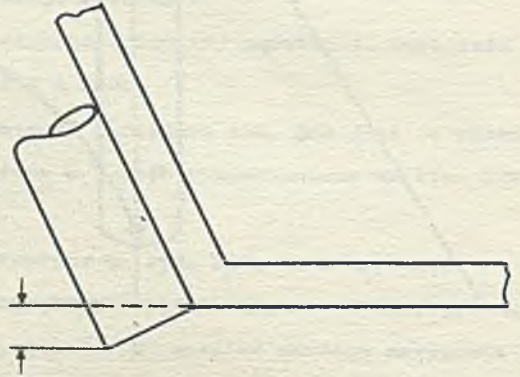
Rys. 226. Ilustracja znaczenia parametrów RIGHT i LEFT /2 parametr/ w instrukcji VTLAXS, opisujących kierunek nachylenia osi narzędzia

- a/ położenie narzędzia przed jego nachyleniem
- b/ nachylenie narzędzia w prawo za pomocą instrukcji VTLAXS/ON, RIGHT, ...
- c/ nachylenie narzędzia w lewo, za pomocą instrukcji VTLAXS/ON, LEFT, ...

PL - płaszczyzna równoległa do płaszczyzny XY

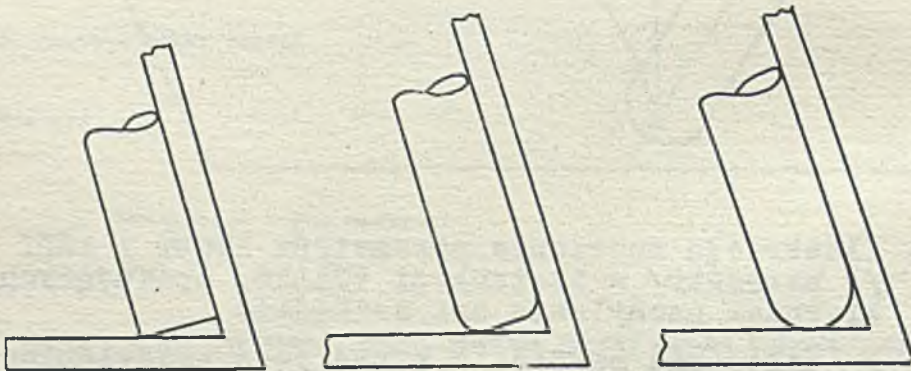


Rys. 228. Ilustracja wykorzystania modyfikatora ZSMALL w instrukcji VTLAXS



Rys. 227. Ilustracja znaczenia w instrukcji VTLAXS modyfikatorów RIGHT i LEFT /6 parametr/ ustawiających narzędzie względem powierzchni /kierunek ruchu narzędzia leży w płaszczyźnie rysunku/

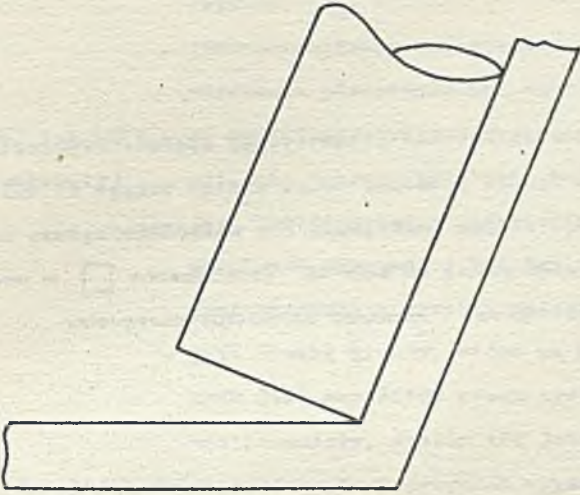
- a/ niemodyfikowane nachylenie osi narzędzia
- b/ ustawienie narzędzia po prawej stronie powierzchni instrukcją
VTLAXS/ON, RIGHT, -, -, -, RIGHT
- c/ ustawienie narzędzia po lewej stronie powierzchni instrukcją
VTLAXS/ON, RIGHT, -, -, -, LEFT



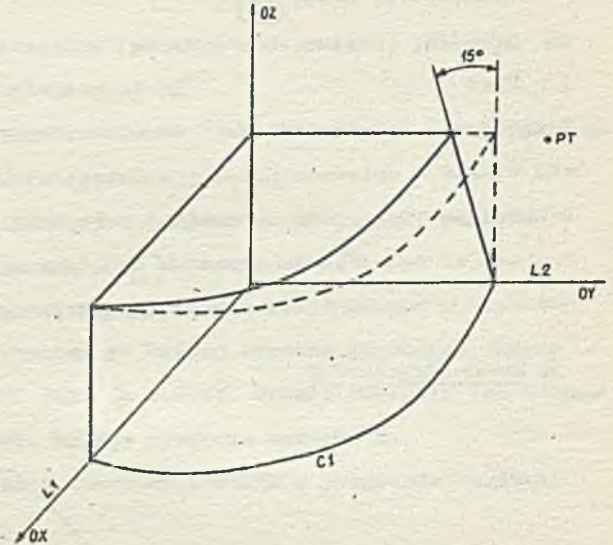
Rys. 229. Ilustracja warunków, w których modyfikator ZSMALL w instrukcji VTLAXS, nie powinien być stosowany

VTLAXS/ON...) i kąt ten podczas przesuwania się narzędzia będzie liniowo zmieniał się aż osiągnie wartość β (podana jako parametr w instrukcji VTLAXS/OFF,...) na końcu drogi narzędzia.

Poniżej przedstawiono przykład prostego programu obróbki części wykorzystującego instrukcję VTLAXS.



Rys. 230. Ilustracja sytuacji, w której modyfikator ZSMALL w instrukcji VTLAXS nie spowoduje prawidłowego ustawienia narzędzia.



Rys. 231. Kształt części, której obróbka jest realizowana przy wykorzystaniu instrukcji VTLAXS, opisany w przykładzie 2 w punkcie 22

Przykład 2

Rys. 231 przedstawia kształt części, którą chcemy uzyskać w wyniku obróbki. Realizuje to poniższy program obróbki części:

```

1 PARTNO PRZYKŁAD
2 PT=POINT/-2,6,1
3 C1=CIRCLE/ 0,0,0,5
4 L1=LINE/XAXIS
5 L2= LINE /YAXIS
6 CUTTER / 1
7 MULTAX
8 TLAXIS / (VECTOR / 0,0,1)
9 FROM/PT
10 G0 / TO, L2,
```



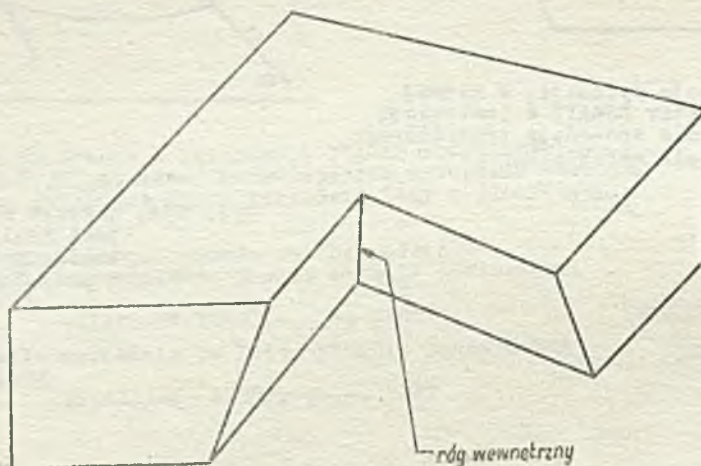
```

11 TIRGT, GORGT/L2, PAST, L1
12 GOLFT/L1, PAST, C1
13 VTLAXS/ON, LEFT, ☐, 1, Ø, RIGHT
14 GORGT/C1, PAST, L2
15 VTLAXS/OFF, LEFT, ☐ 15
16 GOTO/PT
17 FINI

```

W programie tym instrukcje o numerach 2 - 5 definiują kształt geometryczny części. Przekształceniu VTLAXS - opisanemu przez instrukcję 13 i 15 - będzie podlegać droga wzdłuż okręgu C1 opisana instrukcją 15. Podczas przesuwania się wzdłuż tego okręgu nachylenie osi narzędzia będzie zmieniać się od kąta 0° (zaznaczone ☐ w położeniu początkowym) do kąta 15° (zaznaczone ☐ w położeniu końcowym). Instrukcje o numerach 9, 10, 16 określają początkowe i końcowe ustawienie narzędzia.

• Instrukcja WCORN



Rys. 232. Przykład części, przy której wykonaniu można wykorzystać instrukcję WCORN

Instrukcja WCORN może być użyta jedynie w połączeniu z instrukcją VTLAXS, do usunięcia materiału w rogach (rys. 232). Jest ona umieszczona między instrukcją VTLAXS/ON oraz odpowiadającą jej instrukcją VTLAXS/OFF, bezpośrednio po instrukcji ruchu, która powoduje przesunięcie narzędzia w celu wykonania rogu.

Miedzy parą instrukcji VTLAXS/ON i VTLAXS/OFF może pojawić się tylko jedna instrukcja WCORN. Nie stanowi to jednak żadnego ograniczenia, ponieważ w programie obróbki części może pojawić się dowolnie wiele par instrukcji VTLAXS/ON i VTLAXS/OFF, między którymi może oczywiście wystąpić instrukcja WCORN.

Instrukcja WCORN ma następującą postać ogólną:

$$WCORN / \begin{bmatrix} \text{RIGHT} \\ \text{LEFT} \end{bmatrix}, \alpha, \begin{bmatrix} \text{RIGHT} \\ \text{LEFT} \end{bmatrix}, \beta, n, [n]$$

gdzie

RIGHT lub LEFT - są modyfikatorami określającymi początkowy kierunek (w prawo lub w lewo) (1 parametr) względem położenia pionowego, w którym ma być liczony kąt pochylenia osi narzędzia; kierunek ten jest określony, gdy patrzy się zgodnie z kierunkiem ruchu narzędzia.

α - jest początkowym nachyleniem osi narzędzia (podanym w stopniach) liczonym od położenia pionowego; kąt musi być mniejszy od 80°

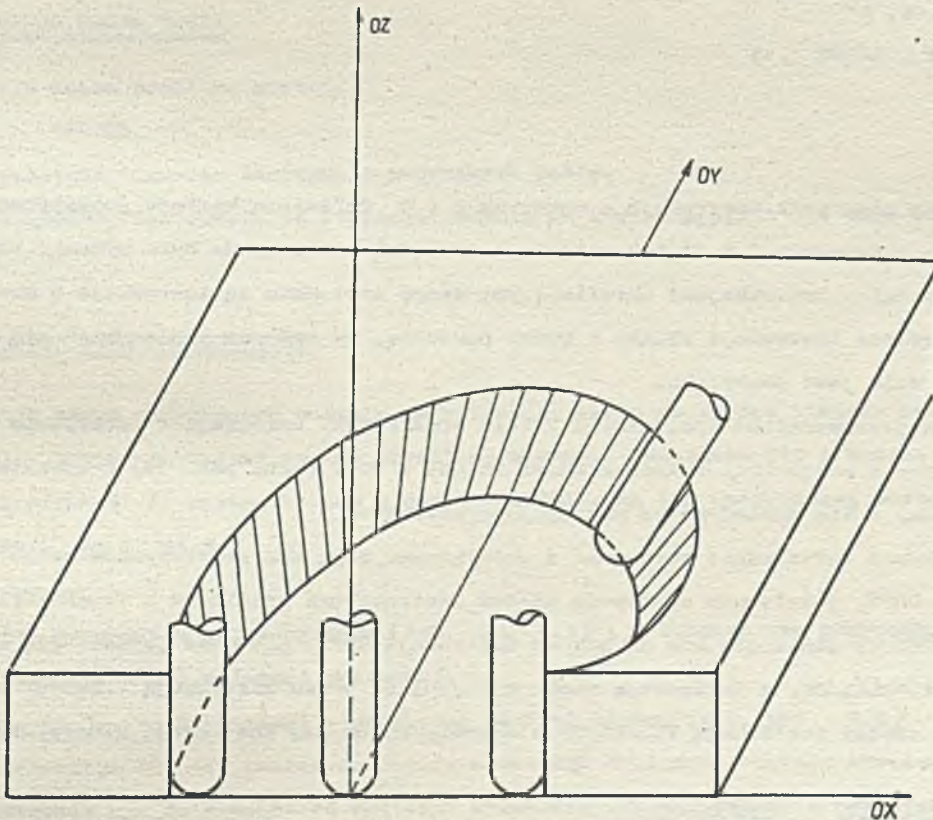
RIGHT lub LEFT - są modyfikatorami mającymi analogiczne znaczenie jak pierwszy parametr instrukcji, ale dotyczą położenia końcowego narzędzia. (3 parametr)

β - jest końcowym kątem nachylenia osi narzędzia (podanym w stopniach) względem położenia pionowego; kąt ten również musi być mniejszy od 80° .

h - jest wysokością warstwy materiału (powierzchni prowadzącej) .

n - jest liczbą kroków, która ma być wykonana po każdej stronie narożnika. Każdy krok jest określony przez system APT jako ≥ 0.0005 . Jeżeli parametr ten zostanie pominięty, system APT jako liczbę kroków przyjmie wartość 6.

Poniżej będzie przedstawiony przykład wykorzystania instrukcji WCORN w programie obróbki części.



Rys. 233. Przykład kształtu części, której obróbka jest realizowana przy wykorzystaniu instrukcji WCORN

Przykład 3

Rys. 233 przedstawia kształt części, którą chcemy otrzymać w wyniku obróbki. Wykonanie tej części można zrealizować za pomocą następującego programu obróbki części.

```

1 PARTNO PRZYKŁAD 3
2 PT=POINT/ 0,0,0
3 C1=CIRCLE /CENTER, -4,4,0 RADIUS, 14
4 C2=CIRCLE /CENTER,6,-7,0, RADIUS,17
5 L1=LINE/0,0,1,0
6 CUTTERX2
7 MULTAX
8 TLAXIS/ (VECTOR/0,0,1)
9 FROM/PT
10 INDIRV /1,0,0
11 GO/TO,C1
12 VTLAXS/ON,LEFT,10,2,.5, LEFT
13 TLLFT, GOLFT/C1, TO,C2
14 WCORN/LEFT, 5., RIGHT , 6 , 1
15 GOLFT/C2, ON, L1
16 VTLAXS/OFF , RIGHT , 15
17 GOTO/PT
18 FINI

```

W powyższym programie instrukcje o numerach 2 + 5 definiują kształt geometryczny części, zaś instrukcje o numerach 9 i 11 tak ustawiają narzędzie, aby można było wykonać skrawanie z pochyleniem narzędzia. Instrukcjami określającymi drogę narzędzia są instrukcje o numerach 13 oraz 15, zaś w programie instrukcje VTLAXS i WCORN powodują, że podczas przesuwania się wzdłuż tej drogi oś narzędzia jest pochylona.

Instrukcje przekształceń specjalnych VTLAXS oraz WCORN, zmieniające nachylenie osi narzędzia, mogą być używane w połączeniu z instrukcjami TRACUT i COPY (zob. pkt 19) przekształcającymi drogę narzędzia, o ile spełnione są następujące warunki:

- instrukcje ruchu definiujące skrawanie z pochyleniem mogą być przetwarzane przez instrukcje TRACUT oraz COPY, jeżeli nie występują między instrukcjami VTLAXS/ON i VTLAXS/OFF
- instrukcja INDEX, odnosząca się do pewnej instrukcji COPY, nie może pojawić się między instrukcjami VTLAX/ON, a instrukcją ruchu występującą przed instrukcją VTLAX/ON nie może także pojawić się między instrukcją VTLAXS/ON a odpowiadającą jej instrukcją VTLAXS/OFF.

23. TWORZENIE PĘTLI W JĘZYKU APT

Język APT umożliwia tworzenie pętli w programie obróbki części, co znacznie zwiększa jego możliwości obliczeniowe a także ułatwia pisanie programów. Pętle pozwalają na iteracyjne definiowanie powierzchni geometrycznych oraz zmiennych skalarnych.

Do zorganizowania pętli wykorzystywana jest instrukcja IF - pozwalająca na wykonanie skoku przy spełnionych określonych warunkach oraz instrukcja skoku bezwarunkowego JUMPTO. Stosowane są również instrukcje LOOPST oraz LOOPND, które umożliwiają wskazanie grupy instrukcji wchodzących w obręb pętli. Instrukcje LOOPST, LOOPND mają jedynie charakter organizacyjny (dla potrzeb systemu APT) tzn. nie powodują żadnych akcji.

Między pojedynczą parą instrukcji LOOPST oraz odpowiadającej jej instrukcji LOOPND mogą się wykonywać dodatkowo inne pętle, zbudowane za pomocą instrukcji skoku warunkowego oraz bezwarunkowego, natomiast wewnątrz grupy instrukcji ograniczonych przez parę LOOPST - LOOPND nie może pojawić się inna para LOOPST - LOOPND.

Poniżej będą dokładniej omówione instrukcje skoków oraz początku i końca pętli.

23.1. Instrukcja początku pętli

Instrukcja początku pętli ma postać:

LOOPST

i musi być pierwszą instrukcją w grupie instrukcji tworzących pętlę. Instrukcja ta jedynie wskazuje na początek pętli i nie daje żadnych informacji wyjściowych.

23.2. Instrukcja końca pętli

Instrukcja końca pętli ma postać:

LOOPND

i musi być ostatnią w grupie instrukcji tworzących pętlę.

Podobnie jak LOOPST, instrukcja ta wskazuje jedynie na koniec pętli i nie daje żadnych informacji wyjściowych.

23.3. Instrukcja skoku warunkowego

Instrukcja skoku warunkowego pozwala na wykonanie skoku do innego miejsca programu obróbki części, o ile został spełniony pewien określony warunek. Działanie jej i budowa jest analogiczna jak dla instrukcji IF arytmetycznej w języku FORTRAN. Ma ona następującą postać:

IF(wa) et1, et 2, et 3

gdzie

wa - jest wyrażeniem arytmetycznym (zob. pkt 4.1) ; w szczególnym przypadku może być ono liczbą lub nazwą zmiennej skalarnej,

et1, et2, et 3 - są etykietami w programie obróbki części (zob. pkt 2.2) .

Działanie instrukcji IF jest takie, że najpierw zostaje obliczona wartość wyrażenia arytmetycznego wa, a następnie - w zależności od wartości wyrażenia arytmetycznego - wykonywany jest jeden z poniższych skoków:

- gdy wartość wa jest mniejsza od zera, wówczas wykonywany jest skok do instrukcji o etykiecie et1,
- gdy wartość wa jest równa zeru, wówczas wykonywany jest skok do instrukcji o etykiecie et2,

• gdy wartość wa jest większa od zera, wówczas wykonywany jest skok do instrukcji o etykiecie et 3.

Należy podkreślić, że instrukcja IF i wszystkie instrukcje z etykietami, do których odwołuje się ta instrukcja muszą występować w obrębie tej samej pętli, tzn. między parą instrukcji $LOOPST$ oraz $LOOPND$ lub w obrębie tej samej makroinstrukcji (zob. pkt 24).

23.4. Instrukcja skoku bezwarunkowego

Instrukcja $JUMPTO$ pozwala na wykonanie skoku bezwarunkowego do dowolnego miejsca programu obróbki części. Ma ona postać:

$JUMPTO/et$

gdzie

et - jest etykietą instrukcji w programie obróbki części.

Po napotkaniu takiej instrukcji, system APT przekazuje sterowanie do instrukcji o etykiecie et . Należy zauważyć, że instrukcja skoku oraz instrukcja z etykietą, do której odwołuje się instrukcja $JUMPTO$, muszą występować w obrębie tej samej pętli lub makroinstrukcji - podobnie, jak to miało miejsce przy instrukcji skoku warunkowego.

23.5. Budowa pętli

Zgodnie z informacjami podanymi we wstępnej części tego punktu, pętlę w języku APT tworzy się za pomocą instrukcji skoku warunkowego (IF) i bezwarunkowego ($JUMPTO$). W celach organizacyjnych wszystkie instrukcje wchodzące w skład pętli powinny zawierać się pomiędzy parą instrukcji $LOOPST$ - $LOOPND$, określających odpowiednio początek i koniec pętli. System APT traktuje pętlę podobnie, jak pojedynczą instrukcję programu obróbki części. Oznacza to, że instrukcje tworzące pętlę są w całości odczytane z pamięci i dopiero wtedy pętla jest przetwarzana. W związku z tym liczba odwołań do pamięci systemu APT jest minimalna tylko wtedy, gdy pętla zawiera jedynie te instrukcje, które są w niej niezbędne. Zminimalizowanie liczby odwołań do pamięci, przez usunięcie poza zakres pętli takich nadmiarowych instrukcji, pozwala na większą efektywność gospodarki pamięcią dla dowolnego programu obróbki części. Powyższe rozważania ilustruje przykład.

Przykład 1

W programie obróbki części zostanie zdefiniowanych 10 prostych przechodzących przez punkt $(\emptyset, \emptyset, \emptyset)$, o zmieniającym się nachyleniu. Przecięcie się tych prostych z prostą $L\emptyset$ tworzy liniowy rozkład punktów. Sytuację tę ilustruje rys. 234.

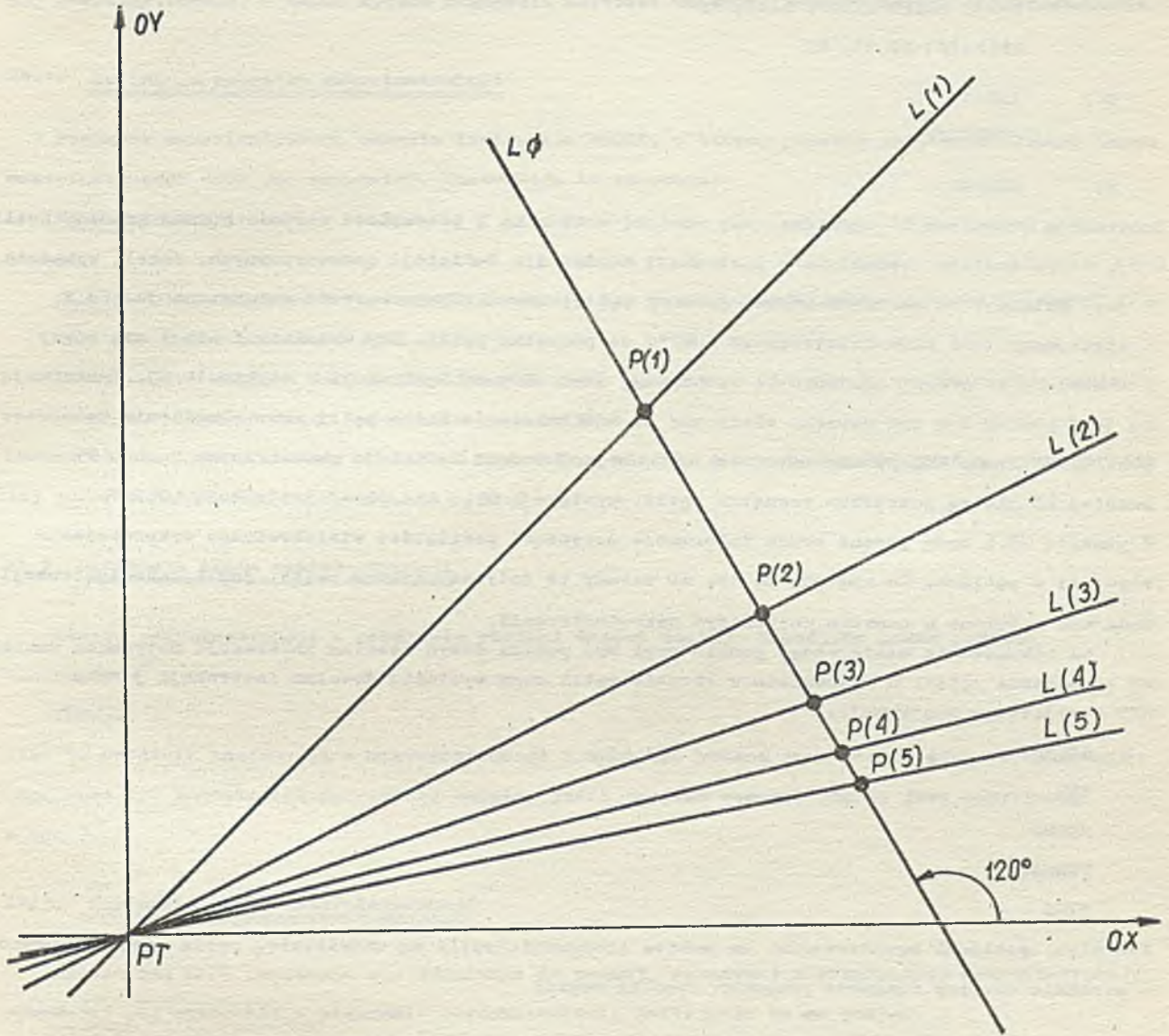
$RESERV/L, 1\emptyset, P, 1\emptyset$

$L\emptyset = LINE / (POINT / 2\emptyset, \emptyset, \emptyset)$, $ATANGL, 12\emptyset$

$PT = POINT / \emptyset, \emptyset, \emptyset$

$I = 1$

$LOOPST$



Rys. 234. Ilustracja graficzna do przykładu 1 w rozdz.23 wykorzystującego instrukcję pętli do definicji geometrycznych.


```

E1)    L(I)= LINE/PT,SLOPE,(1/I)
        P(I)= POINT/INTOF, L(I), L0
        IF(I-10) E2,E3, E3

E2)    I=I+1
        JUMPTO/E1

E3)    LOOPND

```

Instrukcja warunkowa IF sprawdza, czy wartość wskaźnika I przekracza wartość górnej granicy pętli (10), która została określona w instrukcji RESERV dla definicji geometrycznych. Jeżeli wskaźnik I jest mniejszy od wartości górnej granicy pętli, wówczas jego wartość zwiększana jest o 1 i wykonywany jest skok bezwarunkowy JUMPTO do początku pętli. Gdy wskaźnik I staje się równy lub większy od górnej granicy, to wykonywany jest skok do instrukcji o etykiecie E3. Instrukcją tą jest LOOPND i w tym wypadku służy ona do zdefiniowania końca pętli oraz określenia końcowego punktu obliczeń. Należy zauważyć, że instrukcja I=1 oraz definicje geometryczne punktu PT oraz prostej L₀ nie są potrzebne wewnątrz pętli, występują więc one przed instrukcją LOOPST. W punkcie 24.6 będą podane pewne informacje dotyczące możliwości wielokrotnego wykorzystania etykiety w pętlach. Ze względu na to, że zasady te dotyczą zarówno pętli, jak i makroinstrukcji, będą one omówione w punkcie dotyczącym makroinstrukcji.

Na zakończenie niniejszego punktu musi być podana dosyć istotna informacja dotycząca zakresu stosowania pętli, a mianowicie w obrębie pętli mogą wystąpić dowolne instrukcje języka APT z wyjątkiem instrukcji:

```

RESERV
SYN
MACRO
TERMAC
FINI

```

Tak więc, ponieważ ograniczenia na zakres stosowania pętli są niewielkie, pętle mogą obejmować w zasadzie dowolny fragment programu obróbki części.

24. MAKROINSTRUKCJE

Terminem makroinstrukcja określa się w języku APT zbiór instrukcji zawartych między ogranicznikami MACRO i TERMAC, któremu nadano pewną nazwę. Podana nazwa identyfikuje ten zbiór instrukcji i w związku z tym może być on wywołany w dowolnym miejscu programu obróbki części. Odwołanie się do makroinstrukcji następuje za pomocą instrukcji CALL.

Poniżej będą szczegółowo omówione instrukcje MACRO i TERMAC - definiujące odpowiednio początek

i koniec makroinstrukcji oraz instrukcja CALL za pomocą której można odwołać się do zdefiniowanej makroinstrukcji, a także sposób nadawania wartości zmiennym występującym w makroinstrukcji.

24.1. Definicja początku makroinstrukcji

Początek makroinstrukcji określa instrukcja MACRO, w której ponadto jest zdefiniowana nazwa makroinstrukcji oraz jej parametry. Instrukcja ta ma postać:

NAZWA = MACRO/lista

gdzie

NAZWA - jest nazwą makroinstrukcji i podlega tym samym ograniczeniom co nazwa zmiennej (zob. pkt 2.2).

lista - jest ciągiem definicji zmiennych makro (parametry formalne); sposób definiowania zmiennych makro omówiono w punkcie 24.4.

Do makroinstrukcji można się odwoływać w dowolnym miejscu programu obróbki części przez podanie jej nazwy w instrukcji CALL (zob. pkt 24.3).

24.2. Definicja końca makroinstrukcji

Koniec makroinstrukcji w programie obróbki części zostaje określony przez podanie instrukcji:

TERMAC

Jest to ostatnia instrukcja w makroinstrukcji i musi być podana jako jedyne słowo w rekordzie (np. musi być umieszczona na osobnej karcie, jeśli program obróbki części jest wczytywany z kart).

24.3. Odwołanie się do makroinstrukcji

Instrukcją CALL realizuje się odwołanie do pewnej, wcześniej zdefiniowanej makroinstrukcji, powodując jej wykonanie w programie obróbki części. Instrukcja ta ma postać:

CALL/NAZWA,lista

gdzie

NAZWA - jest nazwą wcześniej zdefiniowanej makroinstrukcji,

lista - jest ciągiem definicji lub przeddefiniowań zmiennych makro (zob. pkt 24.4).

Instrukcja ta powoduje wykonanie się makroinstrukcji o podanej nazwie. Przed instrukcją CALL zostaje powtórzony (wewnętrznie zapamiętany) w programie obróbki części fragment programu znajdującej się między instrukcjami:

NAZWA=MACRO/lista

oraz odpowiadającą jej instrukcją

TERMAC

Wystąpienie instrukcji:

CALL/NAZWA, lista

spowoduje wykonanie się makroinstrukcji zgodnie z początkową definicją. W programie obróbki części-

ci można odwoływać się dowolną ilość razy do tej samej makroinstrukcji - przez umieszczenie w programie innych instrukcji CALL.

24.4. Definicje zmiennych makro

Zmienne makro pozwalają na wymianę informacji między podstawowym programem obróbki części a zawartością makroinstrukcji.

Zmienne makro podane są w instrukcji MACRO, definiującej nazwę makroinstrukcji, zaś w instrukcji CALL zostają im przypisane konkretne wartości.

Owołanie się do makroinstrukcji za pomocą instrukcji CALL można traktować jako wstawienie w miejsce instrukcji CALL ciała makroinstrukcji, znajdującego się między ogranicznikami MACRO i TERMAC, przy czym w miejsca nazw zmiennych makro będą wstawione ich konkretne wartości podane w instrukcji CALL. Ilustruje to poniższy przykład.

Przykład 1

Mamy fragment programu obróbki części:

```

.
.
.
MAK1=MACRO/A,B,Z
GO/TO,C1
A,B/C1,ON,L1
GOTO/Ø,Ø,Z
TERMAC

```

E1) CALL/MAK1,A=TLLFT,B=GORGT,Z=Ø

E2) CALL/MAK1,A=TLRGT,B=GOLFT,Z=1Ø

Wykonanie instrukcji o etykiecie E1 oznacza, że w miejsce instrukcji CALL, w wyniku wywołania makroinstrukcji o nazwie MAK1, zostanie wstawiony następujący ciąg instrukcji:

```

GO/TO,C1
TLLFT,GORGT/C1,ON,L1
GOTO/Ø,Ø,Ø

```

natomiast w miejsce instrukcji o etykiecie E2 będzie wstawiony ciąg instrukcji:

```

GO/TO,C1
TLRGT,GOLFT/C1,ON,L1
GOTO/Ø,Ø,1Ø

```

Wywołanie makroinstrukcji o nazwie MAK1 oznacza więc, że zostanie wykonany ciąg instrukcji, które wchodzi w skład tej makroinstrukcji, przy czym w miejsce zmiennych makro będą wstawione konkretne wartości tych zmiennych, podane w instrukcji CALL.

W miejscu zmiennych makro w instrukcji CALL mogą występować:

- nazwy zmiennych lub elementów geometrycznych występujących w programie obróbki części,
- słowa kluczowe języka APT z wyjątkiem słowa CALL oraz słów o ustalonej pozycji,
- liczby.

Nie można natomiast "podstawić" na zmienne makro w instrukcji CALL etykiet lub wyrażeń arytmetycznych.

Dowolna pozycja, reprezentowana przez zmienną makro, może się składać tylko z jednego elementu języka APT (zob. pkt. 2.2. - definicja elementu). Nie może więc wystąpić na jej miejscu np. definicja zagnieżdżona. Należy podkreślić, że zmienna makro może reprezentować zmienną prostą albo zmienną indeksowaną. Jednak w tym drugim wypadku zachodzi konieczność przekazania zarówno nazwy zmiennej jak i indeksu, czyli że muszą być podane dwie zmienne makro. Jedna z nich będzie reprezentować nazwę zmiennej, a druga - jej indeks. Wynika to z wcześniej podanej zasady, że zmienna makro może reprezentować tylko jeden element języka APT, gdyż zarówno nazwa zmiennej jak i indeks (zgodnie z podaną w punkcie 2.2. definicją) są odrębnymi elementami w języku APT. Sposób korzystania ze zmiennych indeksowanych w makroinstrukcji ilustruje podany przykład.

Przykład 2

Rozważmy fragment programu obróbki części.

```
RESERV/TAB,10
MAK2=MACRO/X,I
:
Y=X(I)
:
TERMAC
CALL/MAK2,X=TAB,I=5
N=7
CALL/MAK2,X=TAB,I=N
:
:
```

Przy pierwszym wywołaniu makroinstrukcji MAK2 instrukcja

$Y=X(I)$

wykona się jako instrukcja

$Y=TAB(5)$

natomiast przy drugim wywołaniu makroinstrukcji MAK2 jako instrukcja

$Y=TAB(N)$

Jedynym odstępstwem od reguły, w której zmienna makro może przedstawiać tylko jeden element jest wypadek, gdy wartość jest przedstawiana w postaci liczby ze znakiem np. -7.5. W tym wypadku znak oraz liczba mogą zastępować zmienną makro, a więc kombinacja znaku i liczby może być traktowana jako jeden element. Należy natomiast podkreślić, że zmiennej makro nie może zastępować nazwa zmiennej poprzedzonej znakiem.

Ważną sprawą jest właściwe zrozumienie sposobu, w jaki zmiennym makro przypisuje się ich

wartości aktualne podane przy wywołaniu makroinstrukcji. Polega to na zastąpieniu nazwy zmiennej makro w tekście makroinstrukcji przez wartość aktualną zmiennej makro (podanej po znaku równości w instrukcji CALL) traktowanej jako tekst. Rozważmy to na przykładzie, gdy zmiennej makro nadawana jest wartość liczbowa; nie można jej traktować tak samo, jak zwykłej instrukcji podstawienia arytmetycznego. Pojawienie się nazwy zmiennej makro wewnątrz ciała makroinstrukcji rezerwuje bowiem w instrukcji tylko miejsce, w którym zostaje umieszczony element języka w momencie jego pojawienia się w instrukcji CALL. W chwili wywołania makroinstrukcji zmienna makro jest zastąpiona przez element, który dotychczas reprezentowała. Instrukcja będzie wykonana tak, jakby reprezentowany przez nazwę zmiennej makro element pojawił się wprost w tekście instrukcji. Ilustruje to poniższy przykład.

Przykład 3

Mamy makroinstrukcję

MAK3=MACRO/X

:

X=X+1

:

TERMAC

Przy wywołaniu makroinstrukcji MAK3 w następujący sposób

CALL/MAK3,A

instrukcja

X=X+1

będzie wykonana jako instrukcja

A=A+1

Zauważmy natomiast, że wywołanie opisanej makroinstrukcji MAK3 przez

CALL/MAK3,5

jest błędne, gdyż instrukcja

X=X+1

będzie wykonana jako instrukcja

5=5+1

która jest oczywiście błędna.

Do nazw zmiennych makro stosują się te same zasady co do nazw zmiennych (zob. pkt. 2.2), z wyjątkiem tego, że ta sama nazwa może wystąpić jako nazwa zmiennej makro w więcej niż jednej makroinstrukcji w tym samym programie obróbki części. Poza tym ta sama nazwa może być użyta jako etykieta lub nazwa zmiennej zewnętrznej w stosunku do ciała makroinstrukcji, która wykorzystuje ją jako zmienną makro. Jak już wspomniano zmienna makro nie może być zmienną indeksowaną (przykład 2). Maksymalna liczba zmiennych makro dla dowolnej makroinstrukcji wynosi 50.

Wyróżniamy dwa sposoby nadawania wartości zmiennej makro, tzn. przez określenie tzw. wartości normalnej lub wartości wyznaczonej. Omówmy dokładniej każdy z tych sposobów.

• Wartości wyznaczone zmiennych makro

Zmiennym makro wartości aktualne są nadawane bezpośrednio w instrukcji CALL - przy wywołaniu

makroinstrukcji, jak pokazywano w dotychczasowych przykładach.

Nazwy zmiennych makro występujące w instrukcji CALL muszą być takie same, jak w makroinstrukcji, do której się odwołujemy, natomiast nie muszą one występować w tej samej kolejności. Sposób określenia wartości wyznaczonych zmiennych makro ilustruje poniższy przykład.

Przykład 4

Makroinstrukcję M1 zdefiniowaną jako

M1=MACRO/X,Y,Z

można wywołać w następujący sposób:

CALL/M1, X=1., Y=2.5, Z=-7.6

Zmiennym makro X,Y,Z zostały nadane wartości wyznaczone (odpowiednio 1., 2.5 oraz -7.6) i z takimi wartościami będzie wykonana makroinstrukcja M1.

Wartości wyznaczone mogą być również nadane zmiennym makro, których wartości zostały wcześniej określone (przy definiowaniu makroinstrukcji w instrukcji MACRO) - tzw. wartości normalne. Będzie to dokładniej przedstawione podczas omawiania wartości normalnych zmiennych makro.

• Wartość normalna zmiennej makro

Wartość zmiennej makro można też nadać bezpośrednio w instrukcji MACRO - wówczas nazywamy ją wartością normalną. Jeżeli zmiennej makro nadano wartość normalną, to w instrukcji CALL wywołującej tę makroinstrukcję nie musi się już nadawać tej zmiennej żadnej wartości.

Można jednak zmiennej makro, posiadającej wartość normalną, nadać w instrukcji CALL wartość wyznaczoną i wtedy przy tym wywołaniu, makroinstrukcja będzie realizowana z nadaną w instrukcji CALL wartością zmiennej makro (wartością wyznaczoną). Należy jednak podkreślić, że wartość wyznaczona zmiennej makro zastępuje określoną dla niej wartość normalną tylko przy tych wywołaniach makroinstrukcji, w których wartość wyznaczona pojawia się w instrukcji CALL. Oznacza to, że przy następnych wywołaniach makroinstrukcji, w których nie została podana wartość wyznaczona, obowiązuje wartość normalna zmiennej makro.

Sposób nadawania zmiennym makro wartości normalnych ilustrują poniższe przykłady.

Przykład 5

W makroinstrukcji M2 zdefiniowanej jako

M2=MACRO/A=GORT, B=ALFA,C=5.2

wszystkie zmienne makro (tzn. A,B,C) mają nadane wartości normalne.

Można więc wywołać tę makroinstrukcję przez podanie instrukcji

CALL/M2

Oznacza to, że zmiennej makro nie została w instrukcji CALL nadana inna niż normalna wartość.

Czyli, że przy wykonywaniu makroinstrukcji będą wykorzystane wartości normalne. Makroinstrukcję M2 można też wywołać jako

CALL/M2,A=GOLFT,B=5,C=8

i wówczas będzie ona wykonana z podanymi w instrukcji CALL wartościami wyznaczonymi (choć dla każdej zmiennej makro wcześniej określono wartość normalną).

Z kolei w makroinstrukcji M3, zdefiniowanej jako:

M3=MAKRO/T=5.3,U,W

część zmiennych makro (zmienna T) ma nadaną wartość normalną, natomiast pozostałe zmienne (tzn. U oraz W) nie mają nadanej wartości normalnej. Zmiennym U i W muszą więc być nadane wartości w instrukcji CALL. Natomiast, jeżeli zmiennej T nie będzie nadana inna wartość, to zostanie wykorzystana wartość 5.3.

Sposób określenia wartości zmiennych makro, którym były nadane zarówno wartości normalne, jak i wyznaczone ilustruje poniższy przykład.

Przykład 6

Podczas definiowania makroinstrukcji MAK

MAK=MACRO/X=5,Y,S=ON

nadano wartości normalne zmiennym makro X oraz S. Makroinstrukcję tę można następnie wielokrotnie wywołać w programie obróbki części:

CALL/MAK,X=0,Y=0,S=PAST

CALL/MAK,Y=7

Przy pierwszym wywołaniu makroinstrukcji zostały zdefiniowane wartości normalne zmiennych X oraz S i przy tym wywołaniu makroinstrukcji obowiązują podane w instrukcji CALL wartości wyznaczone. Natomiast przy następnym wywołaniu makroinstrukcji MAK zmienne X i S przyjmują wartości normalne (określone w instrukcji MACRO). Zmiennej Y należy nadawać wartość przy każdym wywołaniu makroinstrukcji MAK, ponieważ nie została dla niej określona wartość normalna.

24.5. Budowa makroinstrukcji

Przy tworzeniu makroinstrukcji należy uwzględnić podane niżej zasady:

- makroinstrukcja nie może być umieszczona (zdefiniowana) wewnątrz innej makroinstrukcji, tzn. między dwiema kolejnymi instrukcjami o postaci

NAZWA=MACRO/lieta

musi wystąpić instrukcja TERMAC

- zmienna makro musi mieć nadaną wartość - albo w postaci wartości normalnej w instrukcji MACRO, albo przez określenie wartości wyznaczonej przy wywołaniu makroinstrukcji, tzn. w instrukcji CALL

- w makroinstrukcji nie mogą wystąpić następujące słowa kluczowe:

RESERV

FINI

SYN

LOOPST

LOOPND

- w programie obróbki części definicja makroinstrukcji musi się pojawić przed instrukcją CALL, do której się ona odwołuje.

Należy podkreślić, że w makroinstrukcji nie mogą pojawić się słowa kluczowe LOOPST oraz LOOPND, ale nie oznacza to, że w makroinstrukcji nie można tworzyć pętli. Rzeczywiście - zgodnie

w wymaganiami języka APT - pętle powinny zawierać się między ograniczającymi je instrukcjami LOOPST oraz LOOPND, ale ten iteracyjny sposób obliczeń jest realizowany za pomocą instrukcji skoku warunkowego oraz bezwarunkowego (IF oraz JUMPTO). Tak więc wewnątrz makroinstrukcji może być realizowany iteracyjny sposób obliczeń - bez względu na ograniczenie dotyczące dopuszczalnych słów kluczowych. Rozważania te ilustruje poniższy przykład.

Przykład 7

Makroinstrukcja DRIL opisuje wiercenie pewnej liczby otworów. Ma ona następującą postać:

DRIL=MACRO/TP,F1,F2,F3,H,N

I=1

E1) GOTO/TP(I), F1

GODLTA/0,0,-H,F2

GODLTA/0,0,H,F3

I=I+1

IF(N-I)E2,E1,E1

E2) TERMAC

CALL/DRIL,TP=P,F1=10,F2=2,F3=5,H=2,N=20

W przedstawionej makroinstrukcji otwory, które należy wywiercić opisane są przez tablicę punktów (zmienna makro TP), zaś liczbę tych punktów określa inna zmienna makro - N. Jako parametr podawana jest również głębokość otworów (zmienna makro H) i szybkość posuwu, z jaką frez będzie przesuwiał się między punktami (F1), zagłębiał się w wykonywany otwór (F2) oraz wycofywał z wykonanego otworu (F3).

Podano również przykład wywołania makroinstrukcji DRIL.

W instrukcji CALL określone wartości zmiennych makro - tak więc zostanie wywierconych 20 otworów określa to zmienna N, przy czym położenie tych otworów jest podane we wcześniej zdefiniowanej tablicy punktów P. Określone są również konkretne wartości szybkości posuwu, z jakimi ma poruszać się narzędzie F1=10 oraz głębokość otworu 2 (zmienna makro H). Warto zwrócić uwagę na sposób, w jaki język APT interpretuje zmienną makro H. W tekście makroinstrukcji występuje instrukcja

GODLTA/0,0,-H,F2

Przy wywołaniu makroinstrukcji i podstawieniu w miejsce zmiennych makro konkretnych wartości, będzie ona miała postać

GODLTA/0,0,-2,2

Tak więc liczba "-2" została jakby złożona ze znaku umieszczonego w tekście makroinstrukcji oraz z cyfry "2", przekazanej za pomocą zmiennej makro H.

Nie jest pożądane umieszczanie wewnątrz makroinstrukcji takich słów języka APT, które - ze swej natury - nie są powtarzalne w programie obróbki części (np. CLPRNT, NOPOST itp.). System APT nie potraktuje tego jako błąd, jednak spowoduje to niepotrzebne kolejne wykonywanie się tych instrukcji.

Wskazane jest również, aby usuwać poza obręb makroinstrukcji wszystkie instrukcje, które nie są niezbędne do jej prawidłowego działania. Wymaganie to spowodowane jest faktem, że wykonywanie makroinstrukcji przez system APT odbywa się w taki sam sposób jak wykonanie pętli, w związku

z tym zmniejszenie liczby instrukcji do koniecznego minimum zmniejsza liczbę odwołań do pamięci.

W celu zademonstrowania sposobu wykorzystywania makroinstrukcji będzie przedstawiony przykład jej zastosowania do definiowania tablicy punktów.

Przykład 8

W postaci makroinstrukcji będzie zapisany przykład 1 z punktu 23.5.

RESERV/L, 1 \emptyset , P, 1 \emptyset

MAKP=MACRO/I, L \emptyset , PT

L(I)=LINE/PT, SLOPE, (1/I)

P(I)=POINT/INTOF, L(I), L \emptyset

TERMAC

PT=POINT/ \emptyset , \emptyset , \emptyset

L \emptyset =LINE/(POINT/2 \emptyset , \emptyset , \emptyset), ATANGL, 12 \emptyset

K=1

LOOPST

E1) CALL/MAKP, K=I, PT=PT, L \emptyset =L \emptyset

K=K+1

IF(K-1 \emptyset)E1, E1, E2

E2) LOOPND

Ten fragment programu realizuje te same funkcje, co fragment programu przedstawiony w punkcie 23.5.

Język APT pozwala na "zanurzanie" makroinstrukcji, tzn.wołanie jednej makroinstrukcji (za pomocą instrukcji CALL) wewnątrz innej makroinstrukcji. Przy czym makroinstrukcjawołana wewnątrz innej makroinstrukcji musi być definiowana na zewnątrz tej makroinstrukcji, np. jeżeli makroinstrukcja MAIX zawiera wewnątrz instrukcję CALL/MAKY, to podczas wykonania makroinstrukcji MAIX (tzn. gdy będzie wykonywana instrukcja CALL/MAIX, lista), będzie też wykonana instrukcja CALL/MAKY. W chwili napotkania tej instrukcji zostanie wykonana makroinstrukcja MAKY, a po niej pozostała część makroinstrukcji MAIX.

Maksymalna liczba poziomów zanurzania makroinstrukcji wynosi 5. Oznacza to, że pierwsza makroinstrukcja woła drugą, druga - trzecią itd., dopóki nie będziewołana piąta makroinstrukcja. Po piątej makroinstrukcji sterowanie musi wrócić do makroinstrukcjiwołanej jako ozwarta, zanim będzie wywołana następna makroinstrukcja. Ilustruje to poniższy przykład.

Przykład 8

W programie obróbki części zostały zdefiniowane następujące makroinstrukcje:

M6=MACRO/....

...

TERMAC

M5=MACRO/....

...

TERMAC


```
M4=MACRO/...
```

```
:
```

```
CALL/M5,...
```

```
:
```

```
CALL/M6,...
```

```
:
```

```
TERMAC
```

```
M3=MACRO/....
```

```
:
```

```
CALL/M4,...
```

```
:
```

```
TERMAC
```

```
M2=MACRO/....
```

```
:
```

```
CALL/M3,....
```

```
:
```

```
TERMAC
```

```
M1=MACRO/...
```

```
:
```

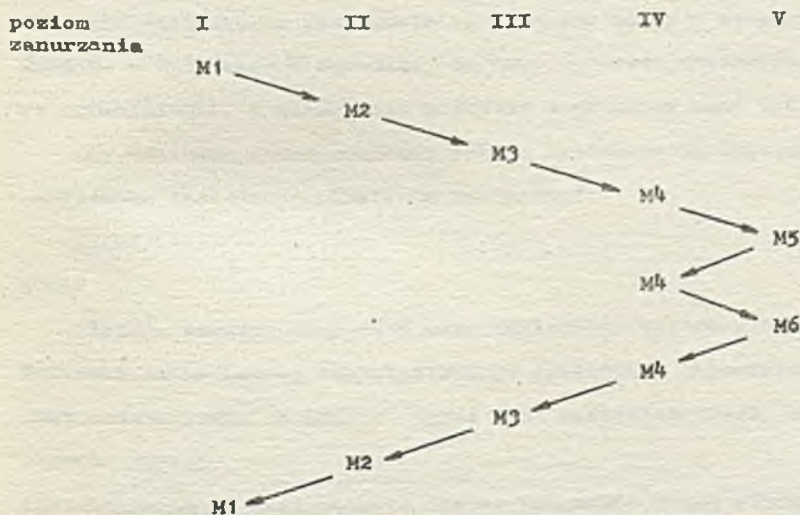
```
CALL/M2,...
```

```
TERMAC
```

W momencie napotkania instrukcji

```
CALL/M1, ....
```

będzie wywołana makroinstrukcja M1, która z kolei woła makroinstrukcję M2, ta z kolei makroinstrukcję M3 itd. Kolejność wykonywania się makroinstrukcji będzie więc następująca:



Należy zauważyć, że makroinstrukcja M5 oraz M6 nie może już wołać innej makroinstrukcji, gdyż zostałaby przekroczona liczba dopuszczalnych poziomów zanurzenia się makroinstrukcji, tzn. 5. Natomiast, gdy po wywołaniu makroinstrukcji M5 w makroinstrukcji M4, sterowanie wróci z powrotem do makroinstrukcji M4, wówczas można wywołać inną makroinstrukcję (tzn. M6). Cyfry rzymskie na schematycznym przedstawieniu kolejności wywoływania makroinstrukcji oznaczają poziom zanurzenia

się makroinstrukcji. W przykładzie nie podano zmiennych makro w instrukcjach MACRO, gdyż nie są one istotne dla przeprowadzonych wyżej rozważań, a mogłyby zaciemniać obraz sytuacji.

24.6. Wielokrotne wykorzystanie etykiety

Nadawanie etykiet instrukcjom języka APT omówiono w punkcie 2.2.

System APT wykrywa błąd, gdy ta sama nazwa etykiety jest używana więcej niż raz w obrębie tej samej pętli lub makroinstrukcji. Natomiast ta sama nazwa wykorzystana jest etykieta w jednej pętli lub makroinstrukcji może być używana w innej pętli lub makroinstrukcji tego samego programu obróbki części. Jest jeden wyjątek od powyższej zasady wielokrotnego wykorzystywania etykiety w różnych pętlach i makroinstrukcjach. Rozważmy sytuację, gdy mamy do czynienia zwołaniem makroinstrukcji z wnętrza innej makroinstrukcji lub też z wnętrza pętli. Załóżmy, że w zewnętrznej makroinstrukcji lub w obszarze pętli występuje etykieta zdefiniowana (tzn. taka, która poprzedza instrukcję a nie taka, do której się odwołujemy). Jeżeli wcześniej nastąpiło odwołanie się do pewnej makroinstrukcji, to ta etykieta nie może być taka sama, jak jakakolwiek etykieta wewnątrz tej makroinstrukcji lub wewnątrz dowolnej innej makroinstrukcji wwołanej z tej makroinstrukcji. Rozważmy następujący przykład.

Przykład 9

Mamy fragment programu obróbki części:

MAK2=MACRO/...

:

E3) I2=I2+1

IF(I2-10)E3,E3,E4

E4) TERMAC

MAK1=MACRO/....

:

E1) I1=I1+1

CALL/MAK2

IF(I1-5)E1,E1,E2

E2) TERMAC

LOOPST

:

E5) I3=I3+1

:

CALL/MAK1,...

:

IF(I3-6)E5,E5,E6

E6) LOOPND

W powyższym fragmencie programu obróbki części żadna z czterech etykiet występujących w makroinstrukcjach MAK1 i MAK2 nie może być identyczna z etykietą E6 zdefiniowaną w obszarze pętli po

odwołaniu się do MAK1. Także żadna z etykiet używanych w makroinstrukcji MAK2 nie może być taka sama, jak etykieta E2 używana w makroinstrukcji MAK1 po odwołaniu się do makroinstrukcji MAK2. Ponadto jedna z etykiet w makroinstrukcjach MAK1 lub MAK2 (lub też zarówno w MAK1 jak i w MAK2) może być taka sama, jak etykieta E5, która jest zdefiniowana przed wywołaniem MAK1. Podobnie każda z dwóch etykiet E3 i E4 używana w MAK2 może być identyczna z etykietą E1 w makroinstrukcji MAK1, ponieważ jest ona zdefiniowana przed odwołaniem się do MAK2.

Ponieważ instrukcje IF i JUMPTO mogą pojawić się tylko w pętli lub makroinstrukcji i mogą odwoływać się tylko do etykiet w tej samej pętli lub makroinstrukcji, to nie występuje niejednoznaczność, jeżeli tę samą etykietę wykorzystujemy wielokrotnie - pod warunkiem, że są spełnione podane wyżej warunki.

Nie zaleca się natomiast stosowania etykiet, do których się wielokrotnie odwołujemy, w instrukcji TRANTO lub w instrukcjach z wielokrotną powierzchnią ograniczającą. W takich sytuacjach system APT przenosi sterowanie do instrukcji, do której ostatnio odwoływano się za pomocą podanej etykiety lub - jeżeli nie wystąpiły takie odwołania - przenosi je do następnej instrukcji zawierającej podaną etykietę. Połączenia instrukcji IF, JUMPTO, wykorzystania wielokrotnej powierzchni ograniczającej oraz TRANTO przy założeniu, że wszystkie korzystają z wielokrotnych etykiet, mogłoby stanowić zbyt skomplikowany problem przy analizie programu obróbki części.

24.7. Makroinstrukcje systemowe

System APT pozwala na tworzenie bibliotek tzw. makroinstrukcji systemowych. Biblioteka makroinstrukcji systemowych jest to zbiór makroinstrukcji zapamiętanych w taki sposób, aby każdy program obróbki części mógł z nich korzystać. Wówczas programista nie musi pisać pewnych fragmentów programu obróbki części, lecz może skorzystać z gotowych makroinstrukcji.

Makroinstrukcjom systemowym są nadawane nazwy w momencie ich zapamiętywania. Ustala się je zgodnie z tymi samymi zasadami, co dowolne nazwy zmiennych w języku APT, występuje tylko dodatkowe ograniczenie, a mianowicie pierwszy znak nazwy musi być literą.

Aby wskazać, które makroinstrukcje systemowe są dostępne w programie obróbki części, należy zastosować instrukcję LIBRY. Ma ona postać:

LIBRY/1 lista

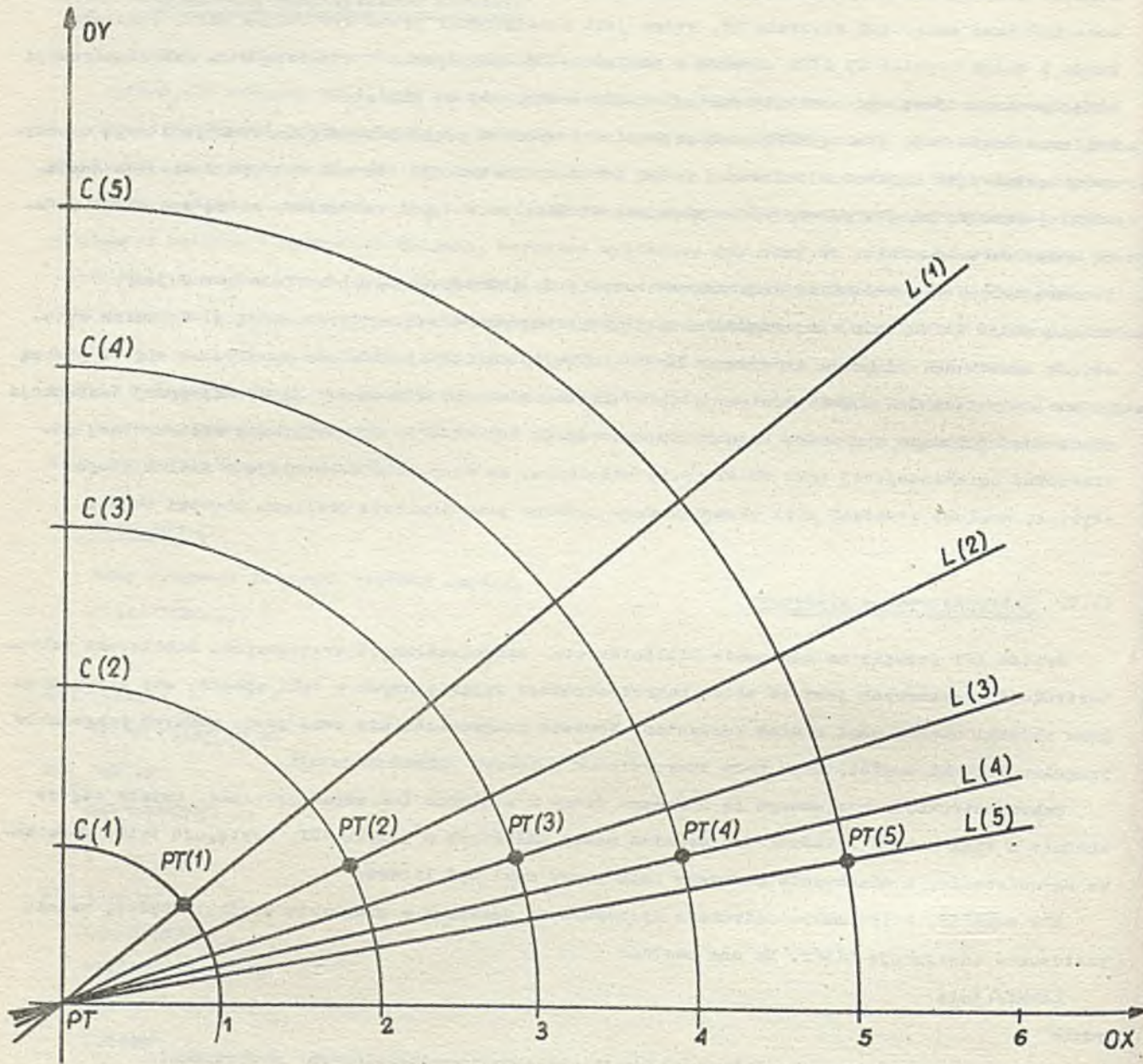
gdzie

lista - oznacza ciąg nazw makroinstrukcji systemowych oddzielonych przecinkami.

Ponieważ nazwa każdej makroinstrukcji systemowej zapamiętana jest w tablicy pośredniej, celowe jest umieszczenie na liście tylko tych makroinstrukcji, które będą używane przez program obróbki części.

Instrukcja LIBRY musi pojawić się w programie przed pierwszym zawołaniem makroinstrukcji systemowej.

W celu przybliżenia problemu będzie przedstawiony przykład wykorzystania makroinstrukcji systemowej.



Rys. 235. Ilustracja graficzna przykładu programu obróbki części wykorzystującego makroinstrukcję systemową do definicji geometrycznych

Przykład 10

Załóżmy, że podana niżej makroinstrukcja została zapamiętana jako makroinstrukcja systemowa

CLM=MACRO/K

$C(K) = \text{CIRCLE}/\phi, \phi, \phi, K$

$L(K) = \text{LINE}/(\text{POINT}/\phi, \phi, \phi), \text{SLOPE}, (1/K)$

$PT(K) = \text{POINT}/\text{XLARGE}, \text{INTOF}, L(K), C(K)$

TERMAC

Makroinstrukcja ta definiuje punkt będący elementem tablicy o nazwie PT, leżącym na przecięciu prostej o nachyleniu $1/K$ oraz okręgu o promieniu K . Ilustruje to rys. 235. Makroinstrukcję tę można wykorzystać w programie obróbki części (do zdefiniowania np. 10 takich punktów) w następujący sposób:

RESERV/L, $1\phi, C, 1\phi, PT, 1\phi$

LIBRY/CLM

I=1

LOOPST

E1) CALL/CLM, K=I

I=I+1

IF(I- 1ϕ)E1, E1, E2

E2) LOOPND

25. PROGRAMY SPECJALNE W JĘZYKU APT

Język APT pozwala programiście na dołączenie programów specjalnych, wykorzystujących techniki programowania, które nie są dostępne w języku APT. Takie programy specjalne nie są częścią systemu APT i muszą być napisane przez użytkownika. Programy te są wołane za pomocą instrukcji CALL. Wyróżniamy dwa sposoby wywołania programu specjalnego, tzn. - przez wartość oraz przez nazwę parametru i przez wartość.

• Wołanie przez wartość

Program specjalny w systemie APT może być wywołany następującą instrukcją:

$$\text{CALL}/\underline{\text{NAZWA}}, \left\{ \begin{array}{l} \text{SEC1} \\ \text{SEC2} \end{array} \right\}, a_1, a_2, a_3, \dots, a_n$$

gdzie:

NAZWA - jest nazwą programu specjalnego, który ma być wykonany; nazwa programu musi się składać z oo najwyżej 6 znaków alfanumerycznych.

SEC1, SEC2 - są słowami w języku APT, które dla systemu APT oznaczają, że program specjalny ma być wykonany odpowiednio w fazie kompilacji lub w fazie pracy jednostki arytmetycznej (zob.pkt. 29).

a_1, a_2, \dots, a_n - są wartościami parametrów potrzebnych w czasie wykonania programu. Maksymalna liczba parametrów wynosi 25. Parametry te muszą być uporządkowane we właściwej kolejności - narzuconej przez program specjalny.

• Wołanie przez nazwę parametru i przez wartość

Instrukcja CALL, wołająca program specjalny z podaniem nazwy parametru i jego wartości, ma postać:

CALL/NAZWA, $\begin{bmatrix} \text{SEC1} \\ \text{SEC2} \end{bmatrix}$, NAZWA1=a1, NAZWA2=a2,...,NAZWAN=an

gdzie

NAZWA, SEC1, SEC2 - mają takie samo znaczenie jak przy wołaniu przez wartość.

NAZWA1, NAZWA2, ..., NAZWAN - są nazwami poszczególnych parametrów w programie specjalnym. Nazwy te mogą się składać z co najwyżej sześciu znaków alfanumerycznych. Parametrów tych może być co najwyżej 25.

Nazwom parametrów NAZWA1, NAZWA2 itd. w instrukcji CALL nadane są wartości odpowiednio a1, a2 itd. Nazwy parametrów NAZWA1, NAZWA2 itd. oraz skojarzone z nimi wartości mogą być uporządkowane w dowolnej kolejności - oczekuje się, że od chwili, gdy parametry zostały nazwane, program specjalny rozróżnia je między sobą.

Nie można łączyć instrukcji wołania przez nazwę i przez wartość tzn. w jednej instrukcji CALL wszystkie parametry muszą być wołane przez wartość lub wszystkie przez nazwę. Nie może część z nich być wołana przez nazwę, a część przez wartość, np. instrukcja

CALL/PROG, SEC2, P1=5., GO.2, P3=27.

Jest niedopuszczalna w języku APT.

Podanie w instrukcji CALL parametru SEC 1 oznacza dla systemu APT, że program specjalny jest dostępny w bibliotece programów APT i będzie wywołany w fazie kompilacji.

Podobnie podanie parametru SEC2 zostanie zdekodowane w fazie kompilacji jako informacja oznaczająca, że program specjalny będzie przetworzony w fazie pracy jednostki arytmetycznej. Nazwy parametrów NAZWA1, NAZWA2 itd. są nazwami programowymi (nadanymi przez programistę) i powinny być różne od wszystkich słów kluczowych języka APT.

26. INSTRUKCJE WEJŚCIA/WYJŚCIA

Instrukcje języka APT mogą inicjować wprowadzanie lub wyprowadzanie danych. Wyprowadzane wyniki (za pomocą instrukcji PRINT lub PUNCH) mogą być wykorzystane przez inne programy obróbki części (jeżeli będą wczytane za pomocą instrukcji - READ). Poniżej zostanie dokładnie omówione działanie instrukcji PRINT, PUNCH oraz READ, a także przedstawione pewne instrukcje specjalne, które umożliwiają wprowadzanie informacji lub wyprowadzanie pewnych wydruków o charakterze informacyjnym (komentarza) w różnych fazach pracy systemu APT (zob. pkt 29).

26.1. Instrukcja PRINT

Instrukcja PRINT pozwala na wydrukowanie lub wyperforowanie jedynie informacji o definicjach geometrycznych lub zmiennych w programie obróbki części. Ma ona następującą postać ogólną:

PRINT/

Ø
1, <u>N1</u> , <u>N2</u> , <u>N3</u> ,
2, <u>N1</u> , <u>N2</u> , <u>N3</u> ,
3, <u>N1</u> , <u>N2</u> , <u>N3</u> ,
3, ALL
4, <u>N1</u> , <u>N2</u> , <u>N3</u> ,
4, ALL

gdzie:

Ø, 1, 2, 3, 4 - określają, w jaki sposób będzie działała instrukcja PRINT (będzie to dokładnie omówione niżej).

N1, N2, N3, - są nazwami zmiennych lub elementów geometrycznych

ALL - jest modyfikatorem oznaczającym, że mają być brane pod uwagę wszystkie elementy geometryczne i zmienne zdefiniowane w programie obróbki części.

Poniżej omówiono działanie każdej z postaci instrukcji PRINT.

● Instrukcja PRINT/Ø

Instrukcja PRINT/Ø powoduje przesunięcie do nowej strony papieru na drukarce. Instrukcja ta używana jest w celu otrzymania ładnej postaci wydruków.

● Instrukcja PRINT/1

Instrukcja w postaci

PRINT/1, N1, N2, N3,

powoduje wydruk w postaci heksadecymalnej zdefiniowanych elementów geometrycznych lub zmiennych o podanych w instrukcji nazwach N1, N2, itd.

● Instrukcja PRINT/2

Instrukcja PRINT/2, N1, N2, N3, ...

powoduje wydruk w postaci kanonicznej podanych wielkości N1, N2, N3, itd. Każda z tych wielkości zostanie wydrukowana od nowej linii, zaś wydruk jej zajmuje jedną lub więcej linii (w zależności od długości postaci kanonicznej tego elementu - zob. pkt 8.1). Wymienione wielkości nie są identyfikowane, tzn. nie są drukowane ich nazwy a jedynie wartości numeryczne związane z tymi wielkościami.

● Instrukcja PRINT/3

Instrukcja ta powoduje wydruk podanych w instrukcji wielkości wraz z ich identyfikatorami.

Jeśli instrukcja została podana w postaci

PRINT/3, N1, N2, N3, ...

wówczas będą wydrukowane podane w instrukcji wielkości o nazwach N1, N2, N3, itd.

Gdy zaś instrukcja wystąpiła w postaci

PRINT/3, ALL

wówczas będą wydrukowane wszystkie aktualnie zdefiniowane zmienne skalarne i elementy geometryczne.

Drukowany identyfikator zmiennej lub elementu geometrycznego składa się:

- z nazwy zmiennej lub elementu geometrycznego,
- indeksu (podanego w nawiasach),
- typu wielkości (tzn. czy jest to zmienna skalarna czy element geometryczny określonego typu),
- liczby parametrów związanych z tą wielkością.

W podany wyżej sposób mogą być drukowane następujące wielkości:

- Liczba (identyfikowana jako skalar),
- zmienna skalarna bez indeksów (identyfikowana jako zmienna skalarna z zerowym indeksem),
- zmienna skalarna indeksowana (identyfikowana jako zmienna skalarna z niezerowym indeksem),
- nazwa elementu geometrycznego bez indeksów (identyfikowana jako nazwa elementu geometrycznego z zerowym indeksem), która nie reprezentuje dużej powierzchni za dużą powierzchnię uważane są: walec tabelaryczny i powierzchnia prostokreślna,
- nazwa elementu geometrycznego z indeksami (identyfikowana jako nazwa elementu geometrycznego z niezerowym indeksem), która nie reprezentuje dużej powierzchni.

Wielkości, które są inaczej określone niż w podany wyżej sposób lub wielkości, które jeszcze nie zostały zdefiniowane, instrukcja PRINT klasyfikuje jako wielkości niezdefiniowane.

• Instrukcja PRINT/4

Instrukcja w tej postaci powoduje wydruk i wyperforowanie podanych wielkości.

Jeżeli instrukcja została podana w postaci:

PRINT/4,N1,N2,N3,....

wówczas będą wydrukowane i wyperforowane wielkości o podanych nazwach N1,N2,N3, itd.

O ile instrukcja została podana w postaci

PRINT/4,ALL

wówczas zostaną wydrukowane i wyperforowane wszystkie aktualnie zdefiniowane zmienne skalarne i elementy geometryczne.

Wydruk zmiennych i elementów geometrycznych następuje w takiej samej postaci, jak opisana przy instrukcji PRINT/3, natomiast perforacja - w formie analogicznej jak opisany przy instrukcji PUNCH/1 (zob. pkt 26.2).

Należy podkreślić, że duże powierzchnie (tzn. powierzchnie prostokątne oraz walce tabelaryczne) będą jedynie wyperforowane w swojej postaci kanonicznej, natomiast nie zostaną wydrukowane.

Zwykle wydruk następuje bezpośrednio po napotkaniu instrukcji PRINT w programie obróbki części. Jeżeli jednak instrukcja PRINT występuje pomiędzy LOOPST-LOOPND, to wydruk następuje dopiero po napotkaniu LOOPND. W wyniku tego można skonstruować cały blok do wydruku (np. w celu tabulacji wydruków) przez umieszczenie instrukcji PRINT w pętli. Ilustruje to poniższy przykład.

Przykład 1

Rozważmy fragment programu obróbki części:

LOOPST

PRINT/Ø

PRINT/3,X,Y,Z,PT(6), L4

PRINT/3,L1,C7

LOOPND

Efektem wykonania tych instrukcji będzie wydruk od nowej strony zmiennych i elementów geometrycznych

nych o nazwach podanych w instrukcjach PRINT. Wydruk nastąpi dopiero po napotkaniu słowa LOOPND. Warto zauważyć, że o ile instrukcje PRINT nie występują między parą instrukcji LOOPST-LOOPND, to wydruk następuje bezpośrednio po napotkaniu każdej instrukcji PRINT. Stąd też wydruk instrukcji programu obróbki części, wyprowadzanej w fazie edycji (zob. pkt.29) i żądanych wyników pośrednich (drukowanych za pomocą instrukcji PRINT) zostaną przemieszane.

• Inna postać instrukcji PRINT

Oprócz omówionych wyżej postaci instrukcji PRINT (powodujących wydruk w momencie ich napotkania w programie obróbki części) istnieje również inna postać instrukcji, a mianowicie:

PRINT/ $\left[\begin{array}{c} \text{ON} \\ \text{OFF} \end{array} \right]$

gdzie:

ON - jest modyfikatorem oznaczającym rozpoczęcie wydruków,

OFF- jest modyfikatorem oznaczającym koniec wydruków.

Instrukcja ta powoduje wydrukowanie postaci kanonicznej niedużych powierzchni (innych niż powierzchnia prostokreślna i walec tabelaryczny) w chwili, gdy tylko zostaną obliczone. Dotyczy to także powierzchni bez nazwy (zdefiniowanych za pomocą definicji zagnieżdżonych). Wydruk następuje w analogicznej formie, jak przy instrukcji PRINT/3. Tak więc będą wydrukowane wszystkie postacie kanoniczne elementów geometrycznych, których definicje znajdują się między parą instrukcji PRINT/ON oraz PRINT/OFF.

26.2. Instrukcja PUNCH

Instrukcja PUNCH pozwala na wyperformowanie w postaci heksadecymalnej postaci kanonicznych zmiennych i elementów geometrycznych (włącznie z dużymi powierzchniami). Wyperformowane wielkości mogą stanowić dane dla innego programu obróbki części, do którego mogą zostać wozytane za pomocą instrukcji READ.

Instrukcja ta ma następującą postać ogólną:

PUNCH/ $\left[\begin{array}{c} 1, \underline{N1}, \underline{N2}, \underline{N3}, \dots \\ 1, \text{ALL} \\ 2, \underline{N1}, \underline{N2}, \underline{N3}, \dots \\ 2, \text{ALL} \\ 3, \underline{N1}, \underline{N2}, \underline{N3}, \dots \\ 3, \text{ALL} \end{array} \right]$

gdzie:

1,2,3 - określają, w jaki sposób będzie działać instrukcja PUNCH (będzie to dokładnie omówione niżej).

$\underline{N1}, \underline{N2}, \underline{N3}, \dots$ - są nazwami zmiennych lub elementów geometrycznych

ALL - jest modyfikatorem oznaczającym, że mają być brane pod uwagę wszystkie elementy geometryczne i zmienne zdefiniowane w programie obróbki części.

Poniżej będzie omówione działanie każdej z postaci instrukcji PUNCH.

● Instrukcja PUNCH/1

Instrukcja w postaci PUNCH/1 powoduje wyperforowanie w formacie heksadecymalnym postaci kanonicznych zmiennych lub elementów geometrycznych (włącznie z dużymi powierzchniami) o podanych w instrukcji nazwach N1, N2, N3 itd. Jeżeli zamiast nazw zmiennych i elementów pojawi się modyfikator ALL, wówczas w formacie heksadecymalnym zostaną wyperforowane wszystkie aktualnie zdefiniowane zmienne i elementy geometryczne.

● Instrukcja PUNCH/2

Instrukcja ta powoduje wyperforowanie w postaci heksadecymalnej zmiennych o nazwach podanych w instrukcji (tzn. N1, N2, N3, itd.) lub wszystkich aktualnie zdefiniowanych zmiennych, o ile w instrukcji wystąpił modyfikator ALL.

● Instrukcja PUNCH/3

Instrukcja ta powoduje wyperforowanie w formacie heksadecymalnym postaci kanonicznych elementów geometrycznych o nazwach podanych w instrukcji PUNCH (tzn. N1, N2, N3, itd.) lub wszystkich aktualnie zdefiniowanych elementów geometrycznych, jeżeli w instrukcji wystąpił modyfikator ALL.

26.3. Instrukcja READ

Instrukcja READ pozwala na przeczytanie i wprowadzenie do programu obróbki części definicji z bloku kart wyperforowanych za pomocą instrukcji PUNCH przez inny program obróbki części. Pozwala ona również na odczytanie i dołączenie do programu uprzednio zapamiętanych instrukcji języka APT. Instrukcja ta ma następującą postać:

READ/ $\left[\begin{array}{l} 1 \\ 1, \underline{N1}, \underline{N2}, \underline{N3}, \dots \\ 2, \underline{NAZWA} \end{array} \right]$

gdzie

1, 2 - określają, w jaki sposób będzie działać instrukcja READ

N1, N2, N3, ... - są nazwami zmiennych lub elementów geometrycznych,

NAZWA - jest nazwą związaną z określonym blokiem instrukcji programu obróbki części, nadaną mu w momencie jego zapamiętywania.

Poniżej zostanie omówione działanie poszczególnych postaci instrukcji READ.

● Instrukcja READ/1

Instrukcja ta powoduje przeczytanie przez system APT wszystkich definicji z bloku kart PUNCH. Blok kart PUNCH jest efektem wykonania instrukcji PUNCH w jednym z poprzednio wykonywanych programów obróbki części. Taki blok kart jest zakończony specjalną kartą ENDPCH, zwaną też kartą transferu, która jest wyperforowana przez instrukcję PUNCH pierwotnego programu obróbki części.

Instrukcja READ/1, w której nie podano nazw wczytywanych elementów powoduje przeczytanie całego

takiego bloku kart PUNCH aż do karty ENDPCH, przy czym zostają zachowane nazwy nadane elementom przez pierwotny program obróbki części. Tak więc nazwy elementów są przeniesione wraz z ich definicjami do programu obróbki części, w którym wystąpiła instrukcja READ/1.

Natomiast gdy w instrukcji READ/1 wystąpią nazwy N1, N2, N3, itd. wówczas pierwszej definicji w bloku kart PUNCH będzie nadana nazwa N1, drugiej N2 itd.

Jeżeli lista nazw jest krótsza niż liczba - definicje zachowają swoje początkowe nazwy nadane im w pierwotnym programie obróbki części. Jeżeli zaś lista nazw jest dłuższa niż liczba definicji dostępnych w bloku kart PUNCH, to zostanie wydrukowany komunikat ostrzegawczy i nadmiarowe nazwy będą niezdefiniowane.

Blok kart PUNCH musi być umieszczony bezpośrednio po instrukcji READ w programie obróbki części. Jeżeli instrukcja READ wystąpiła między parą instrukcji LOOPST-LOOPND, to blok kart PUNCH powinien pojawić się po instrukcji LOOPND. Gdy natomiast instrukcja READ została umieszczona w makroinstrukcji, to odpowiedni blok kart PUNCH powinien pojawić się zaraz po instrukcji CALL.

• Instrukcja READ/2

Instrukcja READ/2 powoduje odczytanie uprzednio zapamiętanych instrukcji programu obróbki części i włączenie ich do programu obróbki części, tworząc w sumie logiczną całość. Instrukcja ta nazywana jest odczytaniem segmentów, jako że jest używana do wczytywania segmentów programu obróbki części, które zostały uprzednio zapamiętane. Podana w instrukcji READ/2 NAZWA jest identyfikatorem związanym z blokiem instrukcji (segmentem) w momencie jego zapamiętywania, nazwa ta może się składać z co najwyżej sześciu znaków alfanumerycznych, z których pierwszy musi być literą.

Nadana segmentowi nazwa pozwala na późniejsze odszukanie właściwego segmentu i włączenie go do odpowiedniego programu obróbki części (w momencie napotkania instrukcji READ zawierającej tę nazwę).

Są pewne ograniczenia nałożone na odczytanie segmentów, a mianowicie:

- instrukcja READ/2 nie może być czytana przez inną instrukcję READ/2,
- instrukcja READ/2 nie może być używana do odczytania przy pierwszym wywołaniu makroinstrukcji systemowej,
- instrukcja READ/2 nie może być używana w pętli lub w makroinstrukcji.

Należy zauważyć, że możliwe jest zapamiętanie wyjścia z instrukcji PUNCH i przeniesienie go do innego programu obróbki części za pomocą instrukcji READ/2. Aby to zrealizować, segment, który ma być odczytany, powinien zawierać instrukcję READ/1, po której następuje blok kart PUNCH.

26.4. Instrukcje specjalne

W języku APT są instrukcje pozwalające na przekazanie informacji do innych faz przetwarzania systemu APT (zob. pkt. 29). Są to takie instrukcje jak TITLES, PPRINT oraz INSERT. Podobnie jak instrukcja PARTO oraz REMARK (omówione w punkcie 20), słowa kluczowe TITLES, PPRINT oraz INSERT muszą się pojawić na karcie w kolumnach od 1 do 6.

● Instrukcja TITLES

Instrukcja TITLES jest używana wyłącznie podczas fazy kompilacji systemu APT. Podana po słowie TITLES informacja alfanumeryczna służy programiście do tworzenia nagłówków dla stabilizowanych informacji wyprowadzanych przez występujące dalej instrukcje PRINT. Po instrukcji TITLES musi występować instrukcja PRINT, w przeciwnym wypadku TITLES nie spowoduje żadnego wydruku.

Instrukcja TITLES ma następującą postać:

$$\text{TITLES} \left[\begin{array}{l} \text{tekst} \\ \text{RIGHT tekst} \end{array} \right]$$

gdzie

tekst - jest ciągiem znaków alfanumerycznych,

RIGHT - jest modyfikatorem oznaczającym, że wydruk tekstu należy przesunąć w prawo.

Wykonanie instrukcji TITLES polega na wydrukowaniu podanego w nawiasach tekstu (o długości do 66 znaków alfanumerycznych). Jeżeli w kolumnach 7 - 12 znajduje się słowo RIGHT poprzedzone znakiem spacji, to tekst zapisany w kolumnach od 13 do 72 zostaje przesunięty w czasie wydruku o 66 znaków w prawo (tzn. będzie wydrukowany począwszy od 67 kolumny).

● Instrukcja PPRINT

Instrukcja ta ma następującą postać:

PPRINT tekst

gdzie

tekst - jest ciągiem do 66 znaków alfanumerycznych, zapisanych na karcie w kolumnach od 7 do 72.

Tekst pojawiający się po słowie kluczowym PPRINT jest przenoszony do późniejszych faz przetwarzania systemu APT - w celu wprowadzenia komentarza przez programistę, np. tekst ten pojawia się na wydruku informacji wprowadzanych do postprocesora w fazie edycji systemu APT (zob. pkt 29) - gdzie została podana instrukcja CLPRNT (zob. pkt 30.3), co umożliwia łatwiejsze odszukanie interesujących programistę wyników.

Instrukcja PPRINT nie powoduje żadnych innych efektów podczas przetwarzania programu obróbki części przez system APT. Może ona pojawić się na listingu postprocesora, co jednak zależy do cech danego postprocesora.

● Instrukcja INSERT

Instrukcja ta ma postać:

INSERT tekst

gdzie

tekst - jest ciągiem do 66 znaków alfanumerycznych podanych na kartach w kolumnach od 7 do 72.

Tekst, który pojawia się w instrukcji po słowie INSERT, jest używany w fazie pracy postprocesora jako informacja, przekazywana bezpośrednio do układu sterowania obrabiarki. W wyniku tego

instrukcja INSERT pozwala na wstawienie ręcznie zakodowanych komend w środek automatycznych obliczeń przeprowadzanych przez postprocesor. Z tego względu należy zwrócić szczególną uwagę na stosowanie instrukcji INSERT.

Niektóre postprocesory używają informacji przekazywanych przez instrukcję INSERT do zasilania urządzenia wyświetlającego obrabiarki, co pozwala operatorowi maszyny na kierowanie procedurami występującymi podczas obróbki części, gdyż jest on wówczas zorientowany, na jakim etapie znajduje się proces obróbki.

27. INSTRUKCJA DEFINICJI POMOCNICZEJ

W języku APT można zdefiniować zmienną skalarną, która przyjmuje wartość określonego parametru postaci kanonicznej podanego elementu geometrycznego. Umożliwia to przekazywanie informacji między elementami geometrycznymi a zmiennymi, jak również na odczytanie wartości poszczególnych parametrów postaci kanonicznej elementów geometrycznych, co może być przydatne np. przy definiowaniu innych elementów geometrycznych.

Instrukcja taka ma postać:

```
OBTAIN, typ/NAZWA, z1, z2, z3, ...
```

gdzie

NAZWA - jest nazwą wcześniej zdefiniowanego elementu geometrycznego,

typ - jest słowem kluczowym określającym typ elementu geometrycznego o podanej w instrukcji nazwie /NAZWA/,

z1, z2, z3, ... - są nazwami zmiennych.

Instrukcja ta definiuje zmienne skalarne o podanych nazwach z1, z2, z3, itd. jak następuje:

z1 przyjmuje wartość pierwszego parametru postaci kanonicznej elementu NAZWA,

z2 przyjmuje wartość drugiego parametru postaci kanonicznej elementu NAZWA,

z3 przyjmuje wartość trzeciego parametru postaci kanonicznej elementu NAZWA, itd.

Podany w instrukcji OBTAIN przecinek (na prawo od NAZWA) określa pozycję (czyli numer parametru) w postaci kanonicznej. Jeżeli więc chcemy odwołać się np. tylko do piątego parametru postaci kanonicznej, wówczas przed podaniem nazwy zmiennej należy umieścić odpowiednią liczbę (w tym wypadku pięć) przecinków. W związku z tym przy określeniu pozycji w postaci kanonicznej należy zwrócić uwagę na podanie właściwej liczby przecinków. Ilustruje to poniższy przykład.

Przykład 1

W programie obróbki części występują instrukcje:

```
C1=CIRCLE/PT1,PT2,PT3
```

```
OBTAIN,CIRCLE/C1,X,Y,Z,,,R
```

Pierwsza instrukcja definiuje okrąg C1 jako okrąg przechodzący przez trzy punkty: PT1, PT2, PT3.

Podana dalej instrukcja OBTAIN pozwala na określenie współrzędnych środka tego okręgu i jego promienia. Podane w tej instrukcji zmienne X, Y, Z oraz R przyjmują wartości:

X - jest współrzędną x środka okręgu C1,

Y - jest współrzędną y środka okręgu C1,

Z - jest współrzędną z środka okręgu C1,

R - jest promieniem okręgu C1.

Podanie trzech dodatkowych przecinków przed parametrem R oznacza, że trzy kolejne parametry postaci kanonicznej okręgu C1 należy pominąć, natomiast zmiennej R należy nadać wartość siódmego parametru postaci kanonicznej czyli promienia okręgu C1.

Liczba parametrów, do których się odwołujemy za pomocą instrukcji OBTAIN nie może przekraczać całkowitej liczby parametrów postaci kanonicznej elementu geometrycznego określonego typu, do którego się odwołujemy (zob. pkt 8.1 - postaci kanoniczne elementów geometrycznych i macierzy). Przy określaniu parametrów instrukcji OBTAIN należy również zwrócić uwagę, aby podany typ odpowiadał rodzajowi elementu geometrycznego NAZWA.

Podane w instrukcji nazwy z1, z2, z3, itd. muszą być nazwami zmiennych.

28. DEFINICJA SYNONIMU

W języku APT słowa kluczowe (zob. pkt 2.2) są słowami zastrzeżonymi. Oznacza to, że programista nie może używać w programie obróbki części nazw identycznych ze słowami kluczowymi. Istnieje natomiast możliwość przemianowywania słów kluczowych (zarówno głównych jak i pomocniczych). Oznacza to, że słowo kluczowe w danym programie obróbki części może występować w dwóch postaciach: w starej ze słownika słów kluczowych języka APT i nowej, nadanej przez programistę. Instrukcja SYN, która umożliwia przemianowywanie słów kluczowych, ma następującą postać:

$$\text{SYN/} \left[\begin{array}{c} \text{słowo, słowo kluczowe} \\ \text{słowo kluczowe, słowo} \end{array} \right] \left[\begin{array}{c} \text{słowo, słowo kluczowe} \\ \text{słowo kluczowe, słowo} \end{array} \right] \left[\dots \right]$$

gdzie

słowo - jest nowym określeniem słowa kluczowego,

słowo kluczowe - jest słowem kluczowym języka APT, które ma być przemianowane.

Instrukcja ta powoduje, że od momentu jej wystąpienia "słowo" jest synonimem "słowa kluczowego", a więc to samo "słowo kluczowe" jest dostępne w programie obróbki części pod dwiema nazwami.

Przykład 1

Podana instrukcja synonimu:

SYN/PT, POINT, C, CIRCLE, PL, PLANE

powoduje, że od momentu jej wystąpienia w programie obróbki części obydwa słowa PT i POINT mogą być używane jako określenie słowa kluczowego POINT. Podobnie C i CIRCLE mogą być użyte dla oznaczenia słowa kluczowego CIRCLE, zaś PL i PLANE dla oznaczenia słowa kluczowego PLANE.

Kolejność podawania słowa kluczowego i jego synonimu w instrukcji SYN określana jest przez pierwszą parę słów (tzn. słowo, słowo kluczowe lub słowo kluczowe, słowo). Ten sposób uporządkowania powinien być zachowany dla następnych par słów.

Przykład 2

Poniższa instrukcja:

SYN/POINT, PT, CIRCLE, C, PLANE, PL

jest równoważna instrukcji podanej w przykładzie 1.

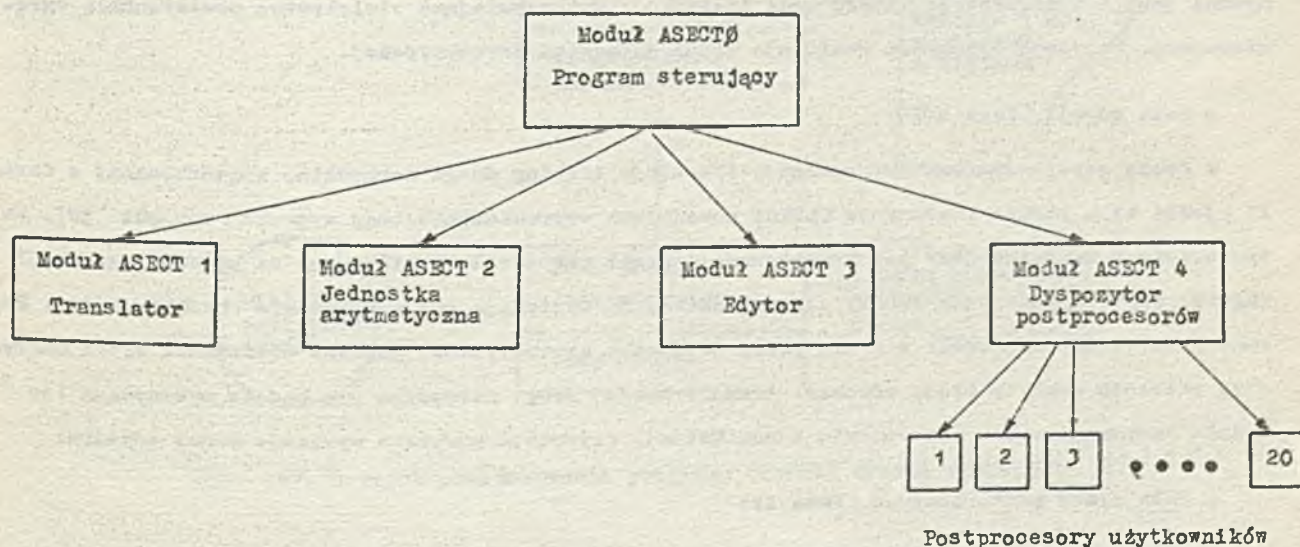
Przy stosowaniu instrukcji SYN należy przestrzegać następujących zasad:

- przemianowanie słowa kluczowego musi wystąpić, zanim w programie obróbki części wystąpi nowe słowo,
- przemianowywanie dotyczy jedynie słów kluczowych,
- żadna z nazw użytych w instrukcji SYN nie może wystąpić jako zmienna makro,
- żadna nazwa podana w instrukcji SYN nie może wystąpić jako nazwa (zmiennej lub elementu geometrycznego) w danym programie obróbki części; może być używana jedynie jako słowo kluczowe, któremu jest przypisana,
- w konkretnym programie obróbki części może być przemianowanych oo najwyżej 50 słów kluczowych.

29. STRUKTURA SYSTEMU APT

Programiście potrzebna jest pewna wiedza o strukturze i działaniu systemu APT, oo m.in. pomaga w zrozumieniu istoty błędów, wykrywanych w różnych fazach przetwarzania systemu. W związku z tym będą podane podstawowe informacje dotyczące pracy samego systemu APT na przykładzie jego konkretnej implementacji na maszynie cyfrowej IBM S/360.

Ogólny schemat struktury systemu APT przedstawia rys. 236.



Rys.236. Struktura systemu APT

System APT składa się z 25 modułów, przy czym 20 modułów przeznaczonych jest na umieszczenie 20 postprocesorów, napisanych przez użytkownika. Pozostałe 5 modułów pełni w systemie następujące funkcje

- ASECT0 - jest modulem sterującym, komunikuje się z systemem operacyjnym oraz wszystkimi pozostałymi modułami systemu APT, które nie mogą się między sobą bezpośrednio komunikować; wykazuje on wszystkie operacje wejścia/wyjścia,
- ASECT1 - jest modulem translatora, który dekoduje instrukcje programu obróbki części,
- ASECT2 - jest modulem jednostki arytmetycznej, który wykonuje potrzebne obliczenia a głównie

oblicza kolejne pozycje narzędzia,

- ASECT3 - jest modulem edytora, który wykonuje wszystkie funkcje związane z redagowaniem, kopiowaniem, drogi narzędzia, wydrukami i kreśleniem drogi narzędzia na autokreślance,
- ASECT4 - jest modulem dyspozytora postprocesorów, który wybiera żądany przez program obróbki części postprocesor i przekazuje sterowanie do modułu zawierającego ten postprocesor.

Napisany przez programistę (zgodnie z regułami języka APT) program obróbki części jest przetwarzany przez poszczególne moduły systemu APT. Można wówczas wyróżnić następujące fazy pracy systemu APT (realizowane przez odpowiednie moduły).

• Faza translacji językowej (Faza I)

W fazie translacji językowej złożone instrukcje języka APT są redukowane do postaci prostego łańcucha elementarnych komend. Są one następnie interpretowane w dalszych fazach pracy systemu APT. Translacja językowa realizowana jest przez moduł translatora.

Podczas translacji językowej wszystkie definicje geometryczne zostają przedstawione w postaci kanonicznej, zaś pętle iwołania makroinstrukcji są zastąpione ich pełnymi postaciami.

• Faza pracy jednostki arytmetycznej (Faza II)

W fazie tej jest obliczana droga narzędzia w postaci położenia punktu końcowego narzędzia (przez podanie współrzędnych x, y, z tego punktu) oraz wektora osi narzędzia. W fazie tej wykonywana jest też instrukcja TRANTO oraz instrukcje wykorzystujące wielokrotną powierzchnię ograniczającą. Powyższe czynności realizuje moduł jednostki arytmetycznej.

• Faza edycji (Faza III)

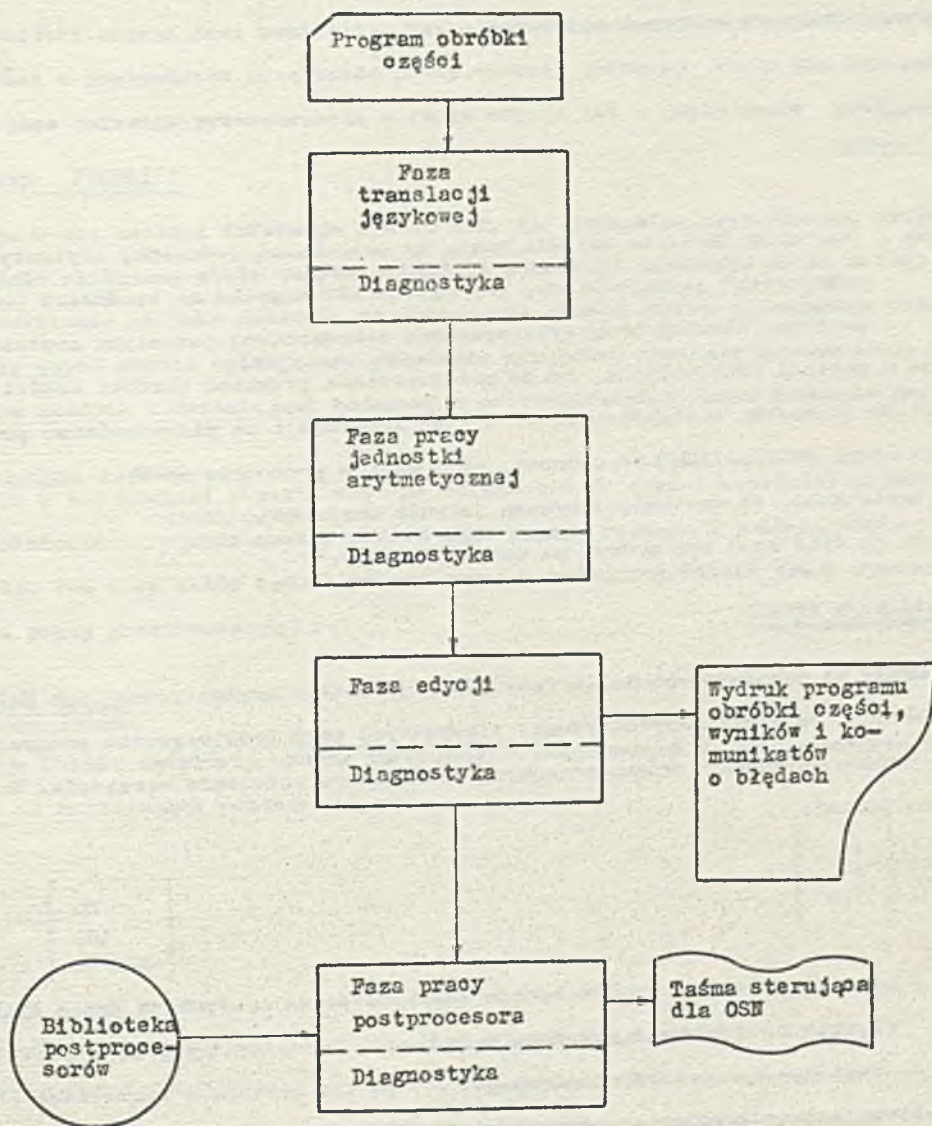
W fazie edycji systemu APT redaguje i drukuje listing drogi narzędzia, wygenerowanej w fazie II (jeśli była podana instrukcja CLPRNT powodująca wyprowadzenie tego wydruku - zob. pkt 30). Na tym etapie dokonywana jest też transformacja drogi narzędzia, realizowana za pomocą instrukcji TRACUT, COPY, VTLAXS oraz WCORN, pod warunkiem, że występują one w programie obróbki części. Należy podkreślić, że jeżeli w fazie pracy jednostki arytmetycznej podczas obliczania drogi narzędzia zostanie wykryty błąd, wówczas transformacja drogi narzędzia nie będzie wykonywana (co będzie zasygnalizowane odpowiednim komunikatem). Czynności powyższe wykonuje moduł edytora.

• Faza pracy postprocesora (Faza IV)

Realizowana jest ona przez moduł dyspozytora postprocesorów, który wybiera żądany w programie obróbki części postprocesor oraz przez moduł wybranego postprocesora. W fazie pracy postprocesora zostają uwzględnione szczególne wymagania układu sterowania obrabiarki sterowanej numerycznie danego typu, jak również generowana jest taśma sterująca dla tej obrabiarki. Odpowiedni zestaw postprocesorów powinien być dołączony do systemu APT przez użytkownika, zaś wybór konkretnego postprocesora dokonywany jest przez podanie jego nazwy w instrukcji MACHIN (zob. pkt 30). Jeżeli w programie obróbki części podano instrukcję NOPOST, wówczas faza pracy postprocesora nie będzie wykonywana.

W każdej z wymienionych wyżej faz pracy każdy z modułów systemu APT realizuje ponadto funkcję diagnostyczną, a mianowicie wykrywa błędy i drukuje odpowiednie komunikaty o błędach. Funkcje

diagnostyczne systemu APT będą szczegółowo omówione w punkcie 33. Proces przetwarzania programu obróbki części w systemie APT można więc schematycznie przedstawić w sposób jak na rys. 237.



Rys. 237. Schemat przetwarzania programu obróbki części w systemie APT

30. INSTRUKCJE STERUJĄCE PRACĄ SYSTEMU APT

Pewne słowa kluczowe języka APT są używane do zdefiniowania sposobu, w jaki system APT realizuje określony program obróbki części. Są to następujące słowa:

CLTV, PTONLY, CLPRNT, NOPLOT, MACHIN, NOPOST oraz FINI.

Znaczenie tych instrukcji będzie omówione w dalszej części tego punktu z wyjątkiem słowa FINI, które już omówiono w punkcie 20.3.

30.1. Instrukcja CLTV

Instrukcja CLTV określa, że zamiast standardowego modułu jednostki arytmetycznej do przetwa-

ranie programu obróbki części należy wykorzystać pomocniczą jednostkę arytmetyczną. Taka jednostka może stanowić część systemu APT, ale powinna być wcześniej dołączona do konkretnej instalacji systemu przez samego użytkownika.

Instrukcja taka ma następującą postać:

CTLV

lub

CTLV/OPTION

gdzie

OPTION - jest modyfikatorem określającym, że pomocniczą jednostkę arytmetyczną należy wykorzystać jedynie wtedy, gdy system APT napotka na trudności podczas przetwarzania programu obróbki przy wykorzystaniu standardowej jednostki arytmetycznej.

Instrukcja w postaci CTLV określa, że do przetwarzania programu obróbki części ma być zastosowana pomocnicza jednostka arytmetyczna. Ta postać instrukcji ma pierwszeństwo przed drugą postacią instrukcji (tzn. CTLV/OPTION) co oznacza, że jeżeli w programie obróbki części wystąpią obie postacie instrukcji, to zostanie wykonana jedynie instrukcja CTLV.

Instrukcja CTLV musi być podana na osobnej karcie.

30.2. Instrukcja PTONLY

Instrukcja ta powoduje ominięcie fazy pracy jednostki arytmetycznej lub dołączenie efektów obliczeń jednostki arytmetycznej otrzymanych przy przetwarzaniu programu obróbki części do innego zbioru (wcześniej wygenerowanego) opisującego położenie narzędzia. Instrukcja ta ma następującą postać:

PTONLY/ $\left[\begin{array}{c} 1 \\ 2 \\ 3, n \end{array} \right]$

gdzie

1, 2, 3 - określają, w jaki sposób będzie interpretowana instrukcja przez system APT (będzie to dokładnie omówione dalej),

n - jest numerem rekordu logicznego.

Poniżej zostaną szczegółowo omówione poszczególne postacie instrukcji PTONLY.

Instrukcja PTONLY powinna pojawić się na oddzielnej karcie lub jako ostatnia pozycja innej instrukcji języka APT.

• Instrukcja PTONLY/1

Instrukcja w tej postaci powoduje ominięcie fazy pracy jednostki arytmetycznej systemu APT. Oznacza to, że podczas przetwarzania programu obróbki części system APT omija fazę pracy jednostki arytmetycznej i bezpośrednio po fazie kompilacji przechodzi do fazy edycji lub fazy pracy postprocesora. Można tak postąpić, gdy dane opisujące położenie narzędzia są zdefiniowane na tyle precyzyjnie, że nie potrzeba dużo obliczeń arytmetycznych i w związku z tym dalsze przetwarzanie programu obróbki części może być realizowane w fazie edycji lub w fazie pracy postprocesora (rys. 237).

● Instrukcja PTONLY/2

Instrukcja w tej postaci powoduje, że faza pracy jednostki arytmetycznej podczas przetwarzania programu obróbki części jest pominięta. Natomiast zbiór danych o nazwie CLTAPE opisujący położenie narzędzia z poprzednich przebiegów pracy maszyny cyfrowej, staje się dostępny dla systemu APT w celu jego dalszego przetworzenia w fazie edycji lub w fazie pracy postprocesora.

● Instrukcja PTONLY/3

Instrukcja w tej postaci informuje system APT, że jednostka arytmetyczna systemu ma wykorzystać uprzednio obliczony zbiór danych opisujący położenie narzędzia aż do n-tego rekordu logicznego i potraktować go jako podstawę do obliczenia nowego zbioru opisującego położenie narzędzia. Pozostała część zbioru opisującego położenie narzędzia zostanie wygenerowana przez jednostkę arytmetyczną zgodnie z definicjami podanymi w przetwarzanym w danym momencie programie obróbki części.

Instrukcja w tej postaci pozwala więc na dołączenie do części wcześniej obliczonego zbioru opisującego położenie narzędzia nowego zbioru tego samego rodzaju i potraktowanie obu tych zbiorów jako całość. Ten nowy zbiór będzie przetwarzany w kolejnych fazach pracy systemu APT (tzn. fazie edycji i pracy postprocesora).

30.3. Instrukcja CLPRNT

Instrukcja CLPRNT umożliwia wydruk informacji wprowadzanych do postprocesora w fazie edycji.

Instrukcja ta ma następującą postać:

CLPRNT

lub

CLPRNT/

ON
OFF
OPTION

gdzie

ON, OFF, OPTION - są modyfikatorami (znaczenie ich będzie omówione dalej).

Pojawienie się instrukcji w postaci CLPRNT (bez modyfikatorów) oznacza, że ma być wydrukowana informacja wprowadzana do postprocesora. Należy zauważyć, że jeśli w programie obróbki części wystąpiła instrukcja transformacji TRACUT (zob. pkt 19.1) to informacja dotycząca położenia punktów drogi narzędzia zostanie wydrukowana zaraz po wykonaniu tego przekształcenia.

Stosowanie instrukcji w postaci CLPRNT/OPTION ma sens, pod warunkiem, że w programie obróbki części wystąpiła instrukcja transformacji (TRACUT). Wówczas drukowana jest informacja o położeniu punktów drogi narzędzia przed wykonaniem przekształcenia TRACUT, przy czym wydruk rozpoczyna się od miejsca, w którym w programie obróbki części wystąpiła instrukcja CLPRNT/OPTION. Ta postać instrukcji CLPRNT ma pierwszeństwo przed innymi postaciami tej instrukcji. Oznacza to, że jeżeli w programie obróbki części wystąpią również inne postacie tej instrukcji (z modyfikatorami), to nie będą one brane pod uwagę.

Podanie instrukcji w postaci CLPRNT/ON powoduje, że informacje wprowadzone do postprocesora będą drukowane dopiero od miejsca, w którym ta instrukcja pojawiła się w programie obróbki części.

Instrukcja w postaci CLPRNT/OFF wskazuje, że w tym miejscu programu obróbki części należy

zakończyć wydruk informacji.

Jeżeli jednak podczas obliczania drogi narzędzia wykryto błąd, wówczas przekształcenie nie zostanie wykonane, a wydrukowane informacje o położeniu punktów drogi narzędzia, będą odnosiły się do sytuacji przed wykonaniem przekształcenia.

30.4. Instrukcja NOPLOT

Instrukcja ta ma jedną postać:

NOPLOT

Pojawienie się jej w programie obróbki części powoduje, że wszystkie instrukcje PLOT/ (zob. pkt 31), jakie pojawiły się w programie, nie zostaną wykonane. Może się ona pojawić w dowolnym miejscu programu obróbki części. Instrukcja ta może występować jako jedyne słowo na karcie lub też w połączeniu z inną instrukcją języka APT. Na przykład pojawienie się w programie:

NOPLOT,CUTTER/5,1

jest dopuszczalne w języku APT i oznacza, że wszystkie instrukcje PLOT/..., które pojawiły się w programie obróbki części nie będą wykonane (instrukcja NOPLOT) oraz, że zostało zdefiniowane narzędzie o średnicy 5 i promieniu zaokrąglenia 1 (instrukcja CUTTER).

30.5. Instrukcja MACHIN

Instrukcja MACHIN określa, który postprocesor ma być wykorzystany do przetwarzania programu obróbki części. Instrukcja ta powinna pojawić się na początku programu obróbki części, aby system APT mógł ją wykorzystać w najbardziej efektywny sposób.

Instrukcja MACHIN ma następującą postać:

MACHIN/NAZWA,lista

gdzie

NAZWA - jest nazwą postprocesora, który ma być wykonywany,

lista - jest ciągiem parametrów oddzielonych przecinkami; liczba i rodzaj parametrów są określone przez konkretny postprocesor.

Podana w instrukcji MACHIN nazwa postprocesora musi się składać ze znaków alfanumerycznych.

W języku APT dopuszczalne są następujące nazwy postprocesorów (są to słowa kluczowe języka):

BENCD	ECS	PIW4E
BENDIX	EXPOST	SC4020
BURG	FOSDIK	SECZRO
CINCY	GECENT	SUNTRN
CONCRD	MILWAK	TRUTRA
DVLIEG	PRAITW	TRW
DYNPAT	PROBOG	

Programista może określić do 9 postprocesorów do wielokrotnego przetwarzania wtórnego w jednym programie obróbki części. Jeżeli w programie obróbki części nie pojawi się żadna instrukcja MACHIN, wówczas system APT przyjmuje, że nie będzie wykonywany żaden postprocesor.

30.6. Instrukcja NOPOST

Instrukcja NOPOST przerywa wykonanie wszystkich postprocesorów określonych w dowolnym miejscu programu obróbki części. Ma ona postać:

NOPOST

Instrukcja ta może wystąpić w dowolnym miejscu programu obróbki części. Może ona być jedynym słowem na karcie lub też występować w połączeniu z innymi instrukcjami języku APT, np. konstrukcja

NOPOST,CUTTER/5,1

jest dopuszczalna w języku APT.

31. STEROWANIE PRACĄ JEDNOSTKI ARYTMETYCZNEJ

Język APT zawiera też pewną liczbę instrukcji, które mają wpływ na sposób przeprowadzania obliczeń przez jednostkę arytmetyczną podczas przetwarzania programu obróbki części. W punkcie tym instrukcje te są omówione szczegółowo. Generalnie pozwalają one:

- na ustalenie trybu pracy jednostki arytmetycznej (instrukcje 2DCALC,3DCALC,NDTEST),
- ograniczenie liczby wektorów drogi narzędzia, składających się na ciąg podstawowych przesunięć (instrukcja NUMPTS),
- określenie maksymalnej długości pojedynczego wektora drogi narzędzia (instrukcja MAXDP),
- wykonanie skoku, opartego na wynikach obliczeń przeprowadzonych przez jednostkę arytmetyczną (instrukcja TRANTO),
- przeprowadzenie obliczeń specjalnych (instrukcja GOUGCK).

Aby właściwie zastosować takie instrukcje, programista powinien znać sposób przeprowadzania obliczeń przez jednostkę arytmetyczną. Niezbędne informacje zostaną podane przy omawianiu poszczególnych instrukcji, ale ze względu na złożoność systemu APT będą to jedynie informacje fragmentaryczne, nie obejmujące całości problemu, a jedynie pewne konkretne sytuacje.

31.1. Ustalanie trybu pracy jednostki arytmetycznej

Najogólniej mówiąc, system APT stosuje technikę obliczeń iteracyjnych w celu określenia wektorów drogi narzędzia, składających się na ciąg podstawowych przesunięć (zob. pkt 14). Wektory obliczane są w taki sposób, aby nie zostały przekroczone zadane tolerancje. System APT ma ponadto możliwość zastosowania rozwiązań analitycznych, co zapewnia dużą efektywność obliczeń. Sam system sprawdza, kiedy można zastosować rozwiązanie analityczne i stosuje je, kiedy tylko jest to możliwe. Metody analityczne mogą być zastosowane, gdy spełnione są poniższe warunki.

• Dla instrukcji ruchu narzędzia tzn. dla instrukcji GOLFT, GORGT, GOFWD, GOUP, GODOWN:

- nie jest używana wielokrotna powierzchnia ograniczająca,
- oś narzędzia jest ustawiona pionowo,
- powierzchnia prowadząca i powierzchnia ograniczająca są płaszczyznami pionowymi lub walcami,
- nie został podany modyfikator PSTAN w instrukcji ruchu,
- powierzchnia przedmiotu nie jest płaszczyzną pionową,

- w definicji narzędzia nie został podany kąt β (zob. pkt. 10) .

• Dla instrukcji przesuwu wstępnego czyli dla instrukcji GO:

- w instrukcji podano co najwyżej dwie powierzchnie,
- nie podano instrukcji THICK dla powierzchni przedmiotu,
- nie podano instrukcji SRFVCT,
- nie podano instrukcji INDIRV lub INDIRP dla powierzchni przedmiotu,
- nie jest podany kąt w definicji narzędzia,
- powierzchnia przedmiotu nie jest płaszczyzną pionową,
- powierzchnia prowadząca jest płaszczyzną pionową lub walcem.

System APT automatycznie sprawdza, czy można zastosować metodę analityczną. Dopiero gdy stwierdzi, że jest to niemożliwe, posługuje się metodami iteracyjnymi. Podanie instrukcji o postaci

3DCALC

powoduje, że w każdym wypadku zostaną zastosowane obliczenia iteracyjne (bez dodatkowego sprawdzania) . Instrukcja ta obowiązuje, dopóki nie będzie unieważniona przez instrukcje o postaci:

2DCALC

lub

NDTEST

Działanie tych instrukcji jest identyczne. Po ich podaniu system APT znowu zaczyna sprawdzać możliwość zastosowania obliczeń analitycznych.

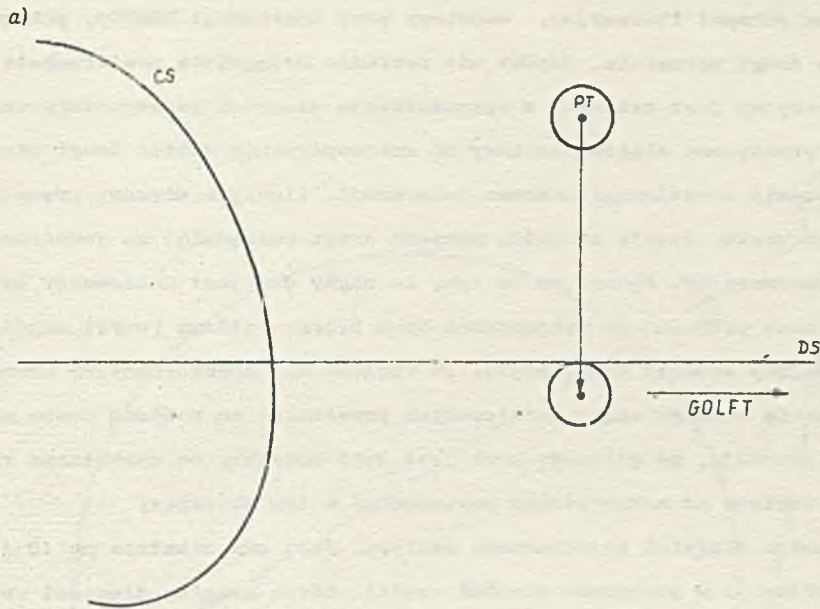
31.2. Instrukcja ograniczająca liczbę punktów

Techniki iteracyjne stosowane przez jednostkę arytmetyczną pozwalają na obliczenie wektora drogi narzędzia, którego początkiem jest położenie narzędzia a następnie sprawdzenie, czy wektor ten można zaakceptować (ma on być tak długi, jak to tylko jest możliwe, aby równocześnie były spełnione wymagania dotyczące tolerancji) . Jeżeli warunek na tolerancję nie jest spełniony, należy obliczyć nowy wektor drogi narzędzia. Czynność tę należy powtarzać tak długo, dopóki nie otrzyma się wektora drogi narzędzia, spełniającego wszystkie wymagania.

Za każdym razem, kiedy zostaje znaleziony kolejny wektor (składający się na ciąg podstawowych przesunięć) spełniający powyższe wymagania, system APT sprawdza, czy będzie osiągnięta powierzchnia ograniczająca. Jeżeli warunek ten jest spełniony, wówczas zostaje zakończona interpretacja danej instrukcji ruchu. Jeżeli natomiast powierzchnia ograniczająca nie została osiągnięta, wówczas, na podstawie położenia końca poprzedniego wektora, znajduje się nowy wektor drogi narzędzia i cała operacja jest powtarzana.

Ogólna koncepcja obliczeń polega na znalezieniu wektora drogi narzędzia, a następnie ciągu podstawowych przesunięć. Należy zauważyć, że takie postępowanie iteracyjne może być zakończone tylko wtedy, gdy zostanie osiągnięta powierzchnia ograniczająca.

Programista powinien liczyć się z faktem, że gdy w programie obróbki części znajdzie się nieprawidłowa instrukcja ruchu, wówczas mogą być wygenerowane wektory drogi narzędzia prowadzące wzdłuż tej samej powierzchni przedmiotu i powierzchni prowadzącej, które nigdy nie osiągną powierzchni ograniczającej (rys. 238).



b/ FROM/PT
GO/PAST, DS
TLLFT, GOLFT/DS, CS

Rys. 238. Przykład podania niewłaściwej instrukcji ruchu narzędzia
a/ ilustracja graficzna
b/ fragment programu obróbki części

System APT zapobiega jednak powstawaniu pętli nieskończonej przez ograniczenie do 400 liczby wektorów drogi narzędzia, tworzących dowolny ciąg podstawowych przesunięć. Jeżeli po obliczeniu 400 wektorów drogi narzędzia wzdłuż podanej pary powierzchni przedmiotu i prowadzącej nie udało się osiągnąć powierzchni ograniczającej, wówczas jest sygnalizowany błąd.

Możliwa jest jednak sytuacja, że należy obliczyć więcej niż 400 wektorów drogi narzędzia aby dotrzeć do podanej w instrukcji powierzchni ograniczającej. Instrukcja NUMPTS powoduje ominięcie ograniczenia do 400 wektorów drogi narzędzia. Ma ona następującą postać:

NUMPTS/n

gdzie

n - jest maksymalną liczbą wektorów drogi narzędzia, składającą się na dowolny ciąg podstawowych przesunięć.

Instrukcja ta obowiązuje, dopóki nie będzie podana nowa instrukcja NUMPTS.

Aby powrócić do standardowych ograniczeń, należy umieścić w programie obróbki części instrukcję: NUMPTS/400.

31.3. Instrukcja określająca maksymalny krok

W ogólnym zarysie generowania ciągu podstawowych przesunięć składa się z dwóch schematów

iteracyjnych. Jeden schemat iteracyjny, omówiony przy instrukcji NUMPTS, polega na obliczeniu kolejnych wektorów drogi narzędzia, dopóki nie zostanie osiągnięta powierzchnia ograniczająca. Drugi schemat iteracyjny jest związany z optymalizacją długości pojedynczego wektora drogi narzędzia. Jednostka arytmetyczna oblicza możliwy do zaakceptowania wektor drogi narzędzia, który nie spowoduje przekroczenia określonego zakresu tolerancji. Algorytm używany przez system APT do obliczenia rozmiaru kroku (czyli długości wektora drogi narzędzia) ma jednocześnie charakter interakcyjny i aproksymacyjny. Polega on na tym, że nigdy nie jest generowany krok przekraczający zadaną tolerancję oraz próbuje się wygenerować krok bliski optimum (czyli najdłuższy możliwy krok nie przekraczający zadanej tolerancji). Ze względu na aproksymacyjny charakter algorytmu, na początku obliczania każdego ciągu podstawowych przesunięć są robione pewne założenia zabezpieczające, które powodują, że pierwszy krok jest dużo mniejszy od normalnego rozmiaru kroku, dopuszczalnego ze względu na zakrzywienie powierzchni w tym obszarze.

Istnieje górna granica długości pojedynczego wektora. Jest ona ustalona na 10 jednostek, jeżeli nie ma innej specyfikacji w programie obróbki części. Górną granicę długości wektora można zmniejszyć za pomocą instrukcji MAXDP. Ma ona następującą postać:

MAXDP/1

gdzie

1 - jest maksymalną długością wektora drogi narzędzia podaną jako liczba jednostek.

Powyższe ograniczenie występuje aż do momentu wystąpienia następnej instrukcji MAXDP.

Aby powrócić do standardowych ograniczeń, należy w programie obróbki części umieścić instrukcję:

MAXDP/10

Instrukcja MAXDP wpływa tylko na obliczenia prowadzone metodą iteracyjną, natomiast nie ma wpływu na obliczenia przeprowadzane metodą analityczną.

Należy podkreślić, że chociaż dla ciągu podstawowych przesunięć, dla którego powierzchnia przedmiotu i powierzchnia ograniczająca są płaszczyznami, jest generowany i wyprowadzany tylko jeden punkt (określający końcowe położenie narzędzia), to jednak wewnętrznie (podczas obliczeń) narzędzie jest przesuwane wielokrotnie o odległość mniejszą lub równą wielkości maksymalnego kroku podanej w MAXDP. A więc dla każdego punktu końcowego może być wewnętrznie generowane (ale nie wyprowadzane) więcej niż jedno przesunięcie. Każde takie przesunięcie jest traktowane jako wektor drogi narzędzia, a licznik punktów w instrukcji NUMPTS (zob. pkt 31.2) jest odpowiednio zwiększany.

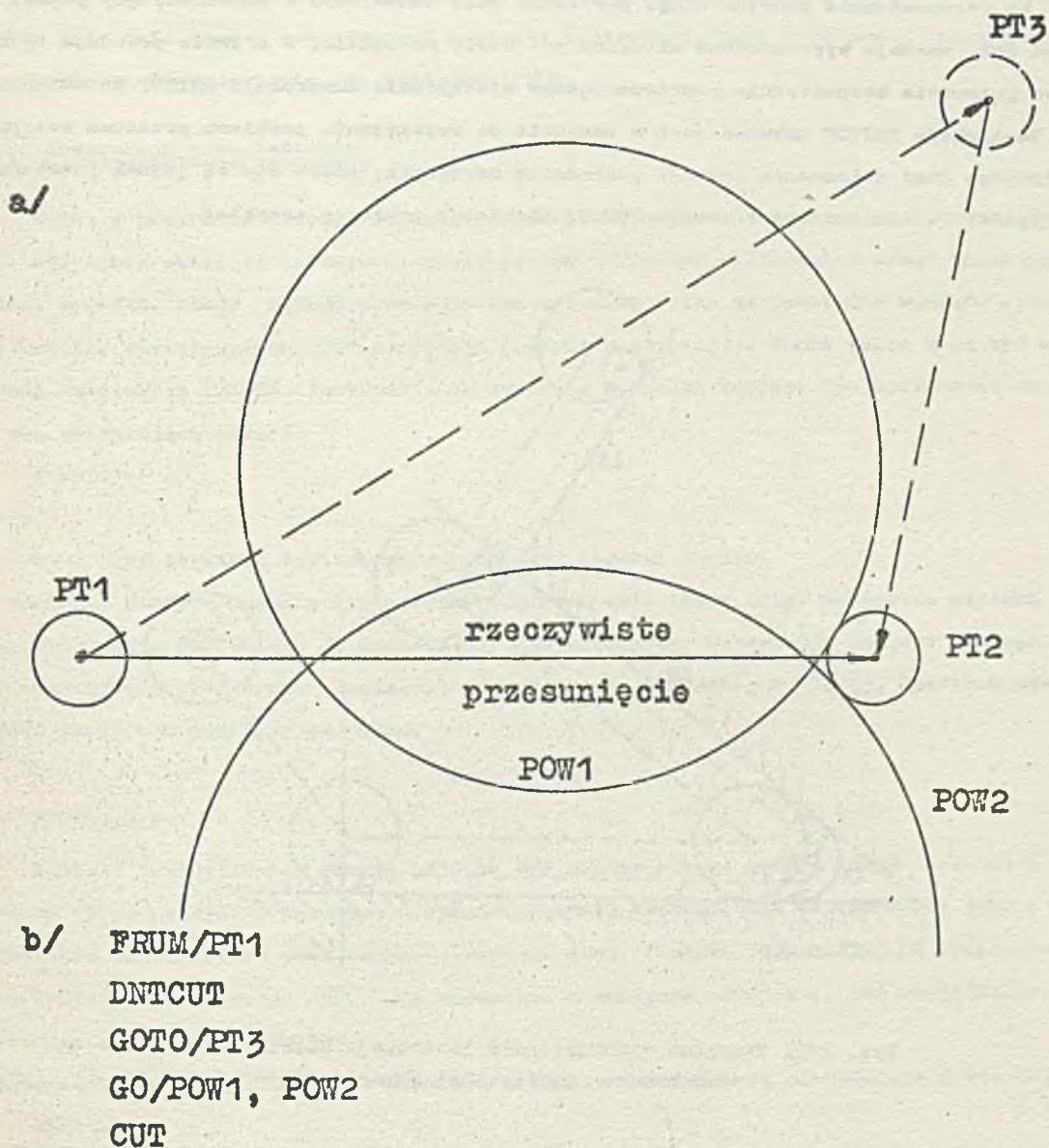
Instrukcja MAXDP ma wpływ jedynie na ciąg podstawowych przesunięć, określony przez instrukcję ruchu narzędzia, natomiast nie ma wpływu na instrukcje programowania punktowego GOTO, GODLTA oraz instrukcje przesuwu wstępnego (GO, OFFSET).

31.4. Instrukcje kasujące wyprowadzanie wyników

Podanie w programie obróbki części instrukcji o postaci:

DNTCUT

powoduje, że następne instrukcje definiujące ruch narzędzia będą wykonywane, ale otrzymane punkty drogi narzędzia nie będą wyprowadzane.



Rys.238 A Przykład wykorzystania instrukcji DNTCUT oraz CUT

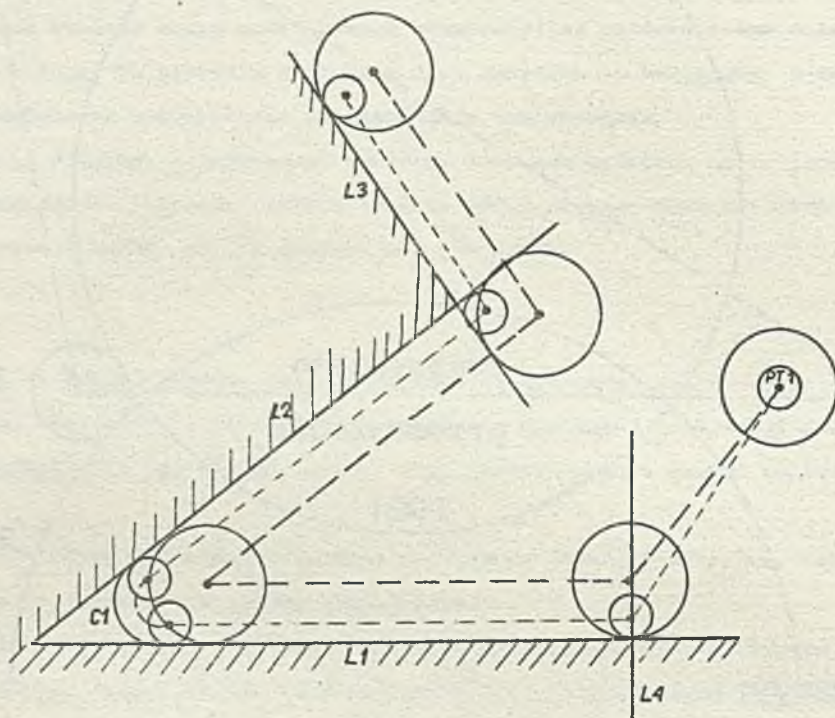
a/ ilustracja graficzna

b/ fragment programu obróbki części

Z kolei podanie instrukcji o postaci:

CUT

powoduje, że wyprowadzanie punktów drogi narzędzia jest wznowione. W momencie, gdy pojawi się instrukcja CUT, zostaje wyprowadzone aktualne położenie narzędzia. W efekcie powoduje to ruch narzędzia od położenia bezpośrednio poprzedzającego wystąpienie instrukcji DNTCUT do aktualnego położenia. Instrukcja DNTCUT używana jest w zasadzie do rozwiązania problemu przesuwu wstępnego, gdzie potrzebne jest obliczenie pozycji pośrednich narzędzia, które nie są jednak potrzebne przy obróbce części. Zastosowanie instrukcji DNTCUT ilustruje poniższy przykład.



Rys. 239. Przykład wykorzystania instrukcji TRANTO i wielokrotnej powierzchni ograniczającej

Przykład 1

Na rys.238a przedstawiono sytuację, w której żądanym położeniem początkowym narzędzia powinien być punkt PT2. W związku z tym narzędzie z punktu PT1 zostaje przesunięte do punktu PT3, a następnie za pomocą zwykłej instrukcji przesuwu wstępnego - przesunięte do punktu PT2. Przesunięcie do punktu PT3 jest poprzedzono instrukcją DNTCUT, natomiast po instrukcji przesuwu wstępnego występuje instrukcja CUT i w związku z tym efekt jest taki, jakby narzędzie było przesuwane bezpośrednio z punktu PT1 do PT2.

Jeżeli program obróbki części pracuje w trybie DNTCUT, to obliczone punkty (z wyjątkiem końcowych) nie pojawiają się w zbiorze wyjściowym i wobec tego nie są dostępne do wynajdywania oraz usuwania błędów w programie obróbki części.

Jeżeli chcemy znać położenie punktów pośrednich, należy podać instrukcję o postaci:

DNTCUT/PRINT

Instrukcja ta powoduje, że będzie prowadzona praca w trybie DNTCUT oraz zostaną wydrukowane punkty pośrednie wraz z wewnętrznymi numerami instrukcji. Taki tryb pracy jednostki arytmetycznej obowiązuje aż do pojawienia się instrukcji CUT.

31.5. Instrukcja przeniesienia

Skoki w programie obróbki części wykonywane są za pomocą instrukcji IF oraz JUMPTO (zob.pkt. 23). Wszystkie skoki są wykonywane przez system APT przed obliczeniem drogi narzędzia. Zachodzą jednak wypadki, kiedy żądany skok może być wykonany tylko na podstawie wyników wykonania instrukcji, określających ruch narzędzia (przykład poniżej). Skoki takie mogą być wykonane za pomocą instrukcji TRANTO. Instrukcja ta zostanie wykonana dopiero po obliczeniu drogi narzędzia. Ma ona następującą postać:

TRANTO/et

gdzie

et - jest etykietą instrukcji w programie obróbki części.

W momencie, kiedy jednostka arytmetyczna podczas obliczania drogi narzędzia napotka taką instrukcję, sterowanie przechodzi do instrukcji o etykiecie et. Instrukcja ta jest szczególnie przydatna, gdy stosuje się wielokrotną powierzchnię ograniczającą (zob.pkt 16.3). Zastosowanie instrukcji TRANTO ilustruje poniższy przykład.

Przykład 2

Kształt przedstawiony na rys. 239 ma być wykonywany za pomocą dwóch narzędzi o różnych rozmiarach (przy przejściu zgrubnym i wykańczającym). Zaokrąglenie zdefiniowane jako C1, wymaga zastosowania wielokrotnej powierzchni ograniczającej. Ponadto ruch wzdłuż L2 wymaga zastosowania modyfikatora kierunkowego GOFWD dla narzędzia o mniejszej średnicy, zaś modyfikatora GORGT dla narzędzia o większej średnicy.

W związku z tym dla narzędzia o większej średnicy potrzebne są następujące instrukcje:

FROM/PT1

GO/L1,PS,ON,L4

TLRGT,GORGT/L1,TO,L2

GORGT/L2,PAST,L3

GOLFT/L3

⋮

natomiast dla narzędzia o mniejszej średnicy:

FROM/PT1

GO/L1,PS,ON,L4

TLRGT,GORGT/L1,TANTO,C1

GOFWD/C1,TANTO,L2

GOFWD/L2,PAST,L3

GOLFT/L3

:

Wykorzystując koncepcję wielokrotnej powierzchni ograniczającej oraz instrukcję TRANTO, te dwa zbiory instrukcji można zapisać razem w następujący sposób:

FROM/PT1

GO/L1,PS,ON,L4

TLRGT,GORGT/L1,TANTO,C1,ET1,TO,L2,ET2

ET1)GOFWD/C1,TANTO,L2

GOFWD/L2,PAST,L3

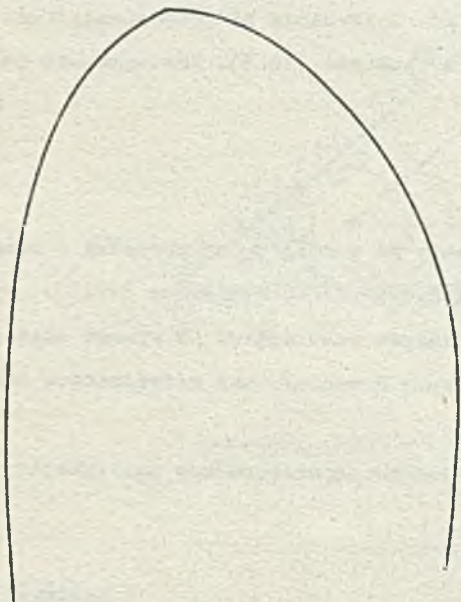
TRANTO/ET3

ET2)GORGT/L2,PAST,L3

ET3)GOLFT/L3

31.6. Instrukcja obliczeń specjalnych

W pewnych warunkach przy przeprowadzaniu obliczeń drogi narzędzia, powierzchnia może mieć ekstremalną zmianę krzywizny na bardzo małym odcinku krzywej. Przykład takiej powierzchni na rys. 240.



Rys. 240. Przykład kształtu powierzchni, przy której może być konieczne wykorzystanie instrukcji GOUGCK

W takim wypadku jednostka arytmetyczna może obliczyć wektor drogi narzędzia, którego punkt końcowy spełnia ograniczenia na tolerancję, natomiast pewne punkty (zwykle położone blisko środka wektora) nie spełniają ograniczeń na tolerancję (rys. 241).

W takich wypadkach należy zastosować instrukcję GOUGCK. Ma ona następującą postać:

GOUGCK / $\begin{bmatrix} \text{ON} \\ \text{OFF} \end{bmatrix}$

gdzie

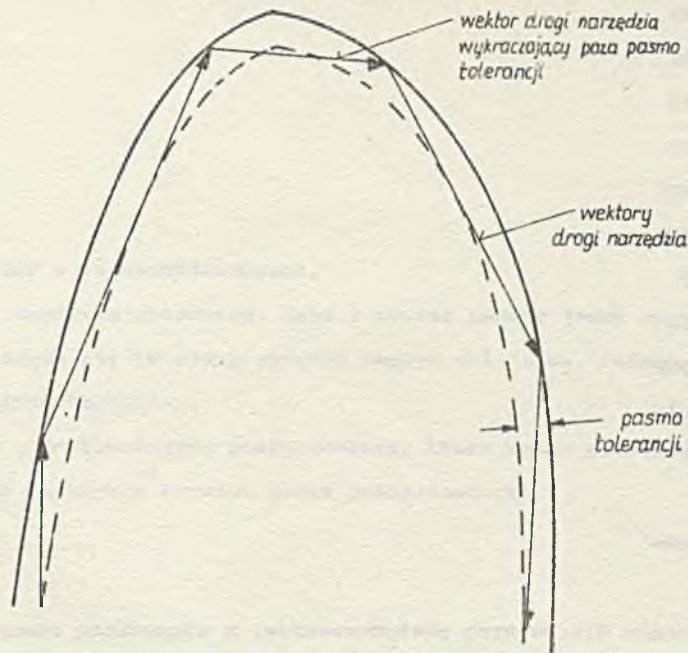
ON - jest modyfikatorem oznaczającym rozpoczęcie obliczeń specjalnych,

OFF - jest modyfikatorem oznaczającym, że należy zakończyć obliczenia specjalne.

Instrukcja GOUGCK/ powoduje, że jednostka arytmetyczna rozpoczyna obliczenia specjalne, przeprowadzane w celu uniknięcia błędów. Obliczenia te kończą się gdy pojawi się instrukcja GOUGCK/OFF. Należy podkreślić, że faza obliczeń specjalnych jest wyjątkowo czasochłonna i powinna być stosowana tylko wtedy, gdy jest to konieczne.

32. INSTRUKCJE POSTPROCESORA

Jak już to było wcześniej sygnalizowane (pkt. 29) postprocesor jest dołączany do systemu APT



Rys. 241. Przykład sytuacji wymagającej zastosowania instrukcji GOUCK

przez jego użytkownika. Zajmuje się on "dopasowaniem" informacji otrzymywanych przy przetwarzaniu programu obróbki części przez system APT do wymagań układu sterowania konkretnej obrabiarki. Zwykle chodzi tu o pewne funkcje obrabiarki pomocnicze w stosunku do samego procesu obróbki np. program obróbki części steruje szybkością posuwu, wybiera szybkość obrotów wrzeciona, włącza i wyłącza dopływ chłodziwa oraz może zmienić narzędzie jeżeli obrabiarka ma takie możliwości. Wszystkie te czynności wykonywane są automatycznie, w trakcie realizacji programu obróbki części przez obrabiarkę.

Powyższe własności mają prawie wszystkie obecnie stosowane układy sterowania obrabiarek. Natomiast nie wszystkie układy rozpoznają słowa postprocesora dostępne w języku APT. W związku z tym konieczne jest jeszcze dodatkowe przetworzenie programu obróbki części przez postprocesor. Postprocesor interpretuje wtedy odpowiednie instrukcje języka APT zgodnie z wymaganiami układu sterowania konkretnej obrabiarki. Ponadto, różne postprocesory mogą też interpretować te same słowa w różny sposób. Dlatego też definicje słów kluczowych postprocesora mają znaczenie tylko w stosunku do wybranego postprocesora. Opisu ich znaczenia należy więc szukać w dokumentacji konkretnego postprocesora.

Poniżej podano spis instrukcji postprocesora dopuszczalnych w języku APT. Przy pewnych słowach kluczowych zostaną również podane pewne zasady, jakie są zwykle stosowane przy określaniu znaczenia tych słów a także postaci całej instrukcji.

32.1. Instrukcje postprocesora bez parametrów

Pewne instrukcje postprocesora występują jako pojedyncze słowa, bez dodatkowych informacji.

Słowami tymi są:

DRAWLI	PROBY
DRESS	PROBX
END	RAPID
FACML	RESET
GOCLER	RETRCT
GONOME	REWIND
ISTOP	STOP
LOCKX	SWITCH
OPSTOP	UNLOCKX
PENUP	UNLOAD
PENDWN	ZERO
PICKUP	

Zwykle słowo kluczowe

END

ma takie samo znaczenie dla różnych postprocesorów, a mianowicie oznacza ono koniec pewnej logicznej części programu. Należy zakończyć wszystkie funkcje pomocnicze, zatrzymać czytnik taśmy i w ten sposób całą obrabiarkę.

Podobnie słowo kluczowe

STOP

oznacza zwykle, że należy zatrzymać czytnik i maszynę w punkcie pozwalającym na zmianę uchwytu, sprawdzenie narzędzia itp.

32.2. Instrukcje postprocesora z parametrami

Pewne instrukcje wymagają dodatkowych informacji, aby mogły być w pełni zrozumiane przez postprocesor. Informacje te przekazywane są za pomocą parametrów. Instrukcja taka ma postać:

SŁOWO/parametry

gdzie

SŁOWO - jest słowem kluczowym języku APT (postprocesora),

parametry - są ciągiem parametrów oddzielonych przecinkami.

Parametrami mogą być zmienne, liczby, nazwy elementów geometrycznych lub słowa kluczowe (modyfikatory). System APT dopuszcza dowolną kombinację parametrów podanych po słowie kluczowym określającym instrukcję postprocesora z parametrami - z wyjątkiem miejsc, gdzie wymagany jest ciąg znaków alfanumerycznych lub występują inne ograniczenia. Np. instrukcja postprocesora może mieć następującą postać:

FEDRAT/f

gdzie f - jest liczbą lub zmienną.

Instrukcja ta definiuje szybkość posuwu (f) podczas obróbki części. Aktualna szybkość posuwu powinna być tak blisko f, jak to jest tylko możliwe, bez przekroczenia dynamicznych ograniczeń obrabiarki (związanych z przeregulowaniem, synchronizacją itp.).

Innym przykładem instrukcji postprocesora z parametrami może być:

COOLNT/ $\left[\begin{array}{c} \text{ON} \\ \text{FLOOD} \\ \text{MIST} \\ \text{TAPKUL} \\ \text{OFF} \end{array} \right]$

gdzie

ON, FLOOD, MIST, TAPKUL, OFF - są modyfikatorami.

Instrukcja ta steruje dopływem chłodziwa. Jeżeli został podany jeden z czterech pierwszych modyfikatorów, wówczas włącza się (w różny sposób) dopływ chłodziwa. Podanie modyfikatora OFF powoduje wyłączenie dopływu chłodziwa.

Poniżej podany jest spis słów kluczowych postprocesora, które muszą wystąpić z parametrami, wraz z listą parametrów, które są zwykle używane przez postprocesory:

AIR/ $\left[\begin{array}{c} \text{ON} \\ \text{OFF} \end{array} \right]$

ARCSLP/ $\left[\begin{array}{c} \text{START, ...} \\ \text{CIRCUL, ...} \end{array} \right]$

ASLOPE/n, ...

AUXFUN/n, $\left[\begin{array}{c} \text{ON} \\ \text{OFF} \end{array} \right]$

CAMERA/ $\left[\begin{array}{c} 1 \\ 2 \\ \text{BOTH} \end{array} \right]$

CHECK/x, y, z, AUTO

CLAMP/n, ...

CLEARP $\left[\begin{array}{l} \text{definicja zagnieżdżona płaszczyzny} \\ \text{nazwa uprzednio zdefiniowanej płaszczyzny} \\ \text{XYPLAN, z} \\ \text{YZPLAN, x} \\ \text{ZXPLAN, y} \end{array} \right]$

CLRSRF/ $\left[\begin{array}{l} \text{nazwa uprzednio zdefiniowanej powierzchni} \\ \text{definicja zagnieżdżonej powierzchni} \\ \text{n, ...} \end{array} \right]$

COOLNT/ $\left[\begin{array}{c} \text{ON} \\ \text{FLOOD} \\ \text{MIST} \\ \text{TAPKUL} \\ \text{OFF} \end{array} \right]$

* COPY/1, $\left[\begin{array}{c} \text{SAME}, n \\ \text{MODIFY}, \underline{MA}, n \\ \text{XYROT}, \alpha, n \\ \text{TRANSL}, x, y, z, n \end{array} \right]$

CORNFED/n, ...

COUPLE/n, ...

CUTCOM/ $\left[\begin{array}{c} \text{LEFT} \\ \text{RIGHT} \\ \text{OFF} \end{array} \right], \left[\begin{array}{c} \text{XYPLAN} \\ \text{YZPLAN} \\ \text{ZXPLAN} \end{array} \right]$

CYCLE/n, ...

DELAY/t

DELETE/ $\left[\begin{array}{c} \text{ON} \\ \text{OFF} \end{array} \right]$

DISPLY/C1, C2, ..., CN

DRAFT/n, ...

DWELL/n, ...

FEDRAT/ $\left[\begin{array}{c} f, \left[\begin{array}{c} \text{AUTO} \\ \text{STEP} \end{array} \right] \\ \text{IPR}, f1, \text{MAXIPM}, f2 \end{array} \right]$

HEAD/ $\left[\begin{array}{c} \text{HIGH} \\ \text{LOW} \\ \text{RIGHT} \\ \text{LEFT} \\ \text{BOTH} \\ n \end{array} \right]$

IFRO/n, ...

* INDEX/1

* INSERT ciąg znaków alfanumerycznych

INTCOD/n

LEADER/n

LETTER/x, y, z, n

LOADTL/n, ...

LPRINT/n, ...

* MACHIN/nazwa, n1, n2, ...

gdzie nazwa - identyfikuje odpowiedni postprocessor

MAGTAP/n, ...

MAXVEL/n, ...

MCHTOL/•

MCHFIN/•

MDEND/n, ...

MDWRIT/n, ...

MODE/n, ...

MOVETO/n, ...

OPSKIP/ $\begin{Bmatrix} \text{ON} \\ \text{OFF} \end{Bmatrix}$

ORIGIN/x,y,z

OVFLOT/n

* PARTNO ciąg znaków alfanumerycznych

PBS/n, ...

PITCH/n, ...

PIVOTZ/z

PLABEL/n, ...

PLOT/ $\begin{Bmatrix} \text{ALL} \\ 1, \dots \end{Bmatrix}$

PLUNGE/d,f

POSTN/ $\begin{Bmatrix} \text{XYPLAN} \\ \text{XZPLAN} \\ \text{ZXPLAN} \end{Bmatrix}$

PLOT/ $\begin{Bmatrix} \text{START, ...} \\ \text{NOMORE} \end{Bmatrix}$

* PRINT ciąg znaków alfanumerycznych

PREFUN/n, ...

REGBRK/n, ...

REVERS/ $\begin{Bmatrix} \text{LEFT} \\ \text{RIGHT} \end{Bmatrix}$, $\begin{Bmatrix} \text{XAXIS} \\ \text{YAXIS} \\ \text{ZAXIS} \end{Bmatrix}$

REWIND/n

$$\text{ROTABL/} \left[\begin{array}{c} 1 \left\{ , \text{ROTREF} \right\} \\ \text{ATANGL}, \alpha \left\{ , \text{ROTREF} \right\} \end{array} \right]$$
ROTHED/ATANGL, C $\left\{ , \text{ROTREF} \right\}$

$$\text{SAFETY/} \left[\begin{array}{c} \text{ON} \\ \text{TURN} \\ \text{FACE} \\ \text{BORE} \end{array} \right]$$

SELCTL/n, ...

SEQNO/n

SLOWDN/n, ...

$$\text{SPINDL/} \left[\begin{array}{c} \text{OFF} \\ \text{rpm} \end{array} \right], \left[\begin{array}{c} \text{CCLW} \\ \text{CLW} \end{array} \right] \dots$$

THREAD/n, ...

THARK/n $\left\{ , \text{AUTO} \right\}$

$$\text{TOOLNO/n, 1,} \left[\begin{array}{c} \text{LARGE} \\ \text{SMALL} \end{array} \right]$$

$$* \text{ TRACUT/} \left[\begin{array}{c} \text{MA} \\ \text{NOMORE} \end{array} \right]$$

TRANS/x, y, z

$$\text{TURRET/n, p, x, y, z,} \left[\begin{array}{c} \text{RAIL} \\ \text{SIDE} \end{array} \right]$$

* VTAXS/n, ...

* VCORN/n, ...

W powyższym spisie pewne instrukcje są wyróżnione gwiazdkami. Znaczenie tych instrukcji było już wcześniej omówione. Są one interpretowane przez system APT oraz przez postprocesor. Należy także podkreślić, że instrukcje:

PARTNO

INSERT

PPRINT

przekazujące informacje alfanumeryczną do postprocesora, muszą pojawić się na karcie w kolumnach od 1 do 6, jak to już było wcześniej sygnalizowane przy omawianiu tych instrukcji.

32.3. Opcjonalna definicja szybkości posuwu

Podane w tym rozdziale słowa postprocesora nie powodują zwykle żadnego działania ze strony podstawowej części systemu APT (z wyjątkiem oznaczonych w spisie gwiazdkami instrukcji przekształceń). Są one po prostu przesyłane do postprocesora, który przeprowadza konieczne do osiągnięcia wymaganego celu obliczenia. Jedynym wyjątkiem od powyższej zasady jest opcjonalna definicja szybkości posuwu.

Ma ona następującą postać

$$\text{FEDRAT} / \begin{pmatrix} fp \\ fp, dm, fm \\ d1, f1 \end{pmatrix}$$

Poniżej zostanie omówione znaczenie poszczególnych postaci tej instrukcji.

Instrukcja

FEDRAT/fp

powoduje ustawienie szybkości posuwu o wartości fp . Szybkość ta powinna być utrzymana, dopóki nie będzie podana nowa wartość szybkości posuwu.

Z kolei instrukcja

$\text{FEDRAT}/fp, dm, fm$

powoduje ustawienie podstawowej szybkości posuwu o wartości fp . Natomiast w odległości dm od każdej powierzchni ograniczającej szybkość posuwu należy zmienić na fm . Po osiągnięciu powierzchni ograniczającej szybkość posuwu przyjmuje z powrotem wartość fp . Instrukcja o postaci

$\text{FEDRAT}/f1, d1$

powoduje, że poprzednio ustalona podstawowa szybkość posuwu będzie obowiązywała, dopóki nie zostanie osiągnięta odległość $d1$ od powierzchni ograniczającej. Od tej chwili szybkość posuwu zmienia się na $f1$. Po osiągnięciu powierzchni ograniczającej zostanie przywrócona podstawowa szybkość posuwu. Instrukcja ta tym się różni od poprzednio omawianej postaci instrukcji, że obowiązuje tylko dla jednej instrukcji ruchu. Oznacza to, że zmiana szybkości posuwu obowiązuje tylko dla jednego, następnego zbliżania się do powierzchni ograniczającej. Następnie system powraca do trybu pracy, w jakim znajdował się poprzednio - czyli określonego przez podanie definicji szybkości posuwu w pierwszej lub drugiej postaci.

Poniżej podano kilka uwag dotyczących stosowania opcjonalnej definicji szybkości posuwu.

Należy zwrócić uwagę, że zmiana szybkości posuwu występuje tylko przy zbliżaniu się do powierzchni ograniczającej, tzn. przy wykonywaniu instrukcji ruchu: GOLFT, GORGT, GOFWD, GOBACK, GOUP, GODOWN. Natomiast przesunięcia spowodowane instrukcjami: GO, GOTO, GODLTA, OFFSET i POCKET nie powodują zmiany szybkości posuwu.

Zdarza się też, że podana w instrukcji FEDRAT odległość od powierzchni ograniczającej ($d1$ lub dm) może być większa niż długość przesunięcia spowodowanego instrukcją ruchu. Jeżeli taka sytuacja wystąpi, wówczas system APT zmieni szybkość posuwu przed wykonaniem instrukcji ruchu

i w związku z tym całe przesunięcie zdefiniowane przez tę instrukcję ruchu będzie odbywać się z szybkością posuwu f_m lub f_1 . W sytuacji takiej nie jest podawany żaden komunikat diagnostyczny.

Należy zauważyć, że podstawowa szybkość posuwu f_p może być określona albo za pomocą instrukcji FEDRAT albo też przez podanie szybkości posuwu w instrukcjach FROM, GOTO, GODLTA lub w instrukcjach ruchu (punkty 13.1, 13.2, 13.3, 16). Podanie powyższej definicji szybkości posuwu powoduje, że od tej chwili szybkość posuwu przy zbliżaniu się do powierzchni ograniczającej nie będzie zmieniana.

Należy podkreślić, że odległość od powierzchni ograniczającej, przy której zmieniana jest szybkość posuwu powinna być dosyć mała - nie większa niż długość 70 wektorów drogi narzędzia (lub 35, jeżeli została podana instrukcja MULTAX), wymaganych do osiągnięcia powierzchni ograniczającej po zmianie szybkości posuwu. Jeśli warunek ten nie jest spełniony, wówczas zmiana szybkości posuwu nastąpi bliżej powierzchni ograniczającej, niż to było określone w instrukcji. Nie jest wówczas podawany żaden komunikat diagnostyczny.

33. DIAGNOSTYKA W SYSTEMIE APT

System APT pomaga w zlokalizowaniu błędów napotkanych podczas przetwarzania programu obróbki części. Każdy z modułów systemu APT (zob. pkt 29) wykrywa i sygnalizuje programiście błędy w programie obróbki części. Z każdym z błędów związany jest pewien numer oraz komunikat, podawany przez odpowiedni moduł systemu, określający przyczynę błędu i ewentualnie sposób jego poprawiania. W niniejszym opracowaniu podany jest opis błędów wykrywanych w fazie translacji (Dodatek B), spis błędów wykrywanych w fazie pracy jednostki arytmetycznej (Dodatek C), oraz spis błędów wykrywanych w fazie edycji (Dodatek D). Błędy wykrywane przez postprocesor powinny być opisane w dokumentacji odpowiedniego postprocesora.

Numer błędu oraz związany z nim komunikat są drukowane w tekście programu w chwili napotkania tego błędu. Tak więc błędy wykrywane w fazie translacji będą drukowane w tekście programu obróbki części, zaś błędy wykrywane w fazie pracy jednostki arytmetycznej i w fazie edycji pojawiają się przy wydruku informacji o drodze narzędzia obliczonej przez program obróbki części.

Oprócz omawianych wyżej błędów stanowiących wyraźne naruszenie składni lub zasad pisania programu w języku APT w programach obróbki części mogą też wystąpić błędy innego rodzaju: a więc przepełnienie różnych tablic systemowych (tablicy synonimów, tablic VST i LTPP opisanych w punkcie 33.5) lub też błędy spowodowane podaniem niewłaściwych danych, których system APT w żaden sposób nie może sprawdzić. Na przykład programista może dostarczyć w pełni akceptowalne przez system dane opisujące powierzchnie, które to dane różnią się jednak o kilka tysięcznych od wartości, jakie są wymagane. Ten rodzaj błędu nie może być wykryty przez system APT - można je jednak wykryć przez porównanie wyników generowanych z pierwszej fazy pracy systemu z danymi podanymi przez programistę. Jedyną możliwością określenia wartości generowanych przez pierwszą fazę pracy systemu APT jest zastosowanie instrukcji PRINT (opisanej w punkcie 26.1), za pomocą której można wydrukować wszystkie lub tylko niektóre postacie kanoniczne powierzchni.

System APT ma też pewne możliwości wspomagania programisty przy wykrywaniu trudnych do zlokalizowania błędów. Poniżej zostaną omówione instrukcje DEBUG oraz DYNDMP, pozwalające na wydruk dodatkowych informacji w fazie translacji lub w fazie pracy jednostki arytmetycznej. Ponadto

będzie omówiona instrukcja TUNEUP pozwalająca na zmianę wielkości tablic systemowych, będą podane pewne dodatkowe informacje na temat podawania definicji w postaci kanonicznej oraz interpretacji definicji tabelarycznego walca, co pozwoli programiście na uniknięcie pewnych błędów.

33.1. Definiowanie elementów geometrycznych w postaci kanonicznej

W punkcie 8.2 omówiono instrukcję CANON, pozwalającą na definiowanie elementów geometrycznych w postaci kanonicznej. System APT przyjmuje dowolne numeryczne dane wejściowe. Aby jednak uzyskać poprawne wyniki, programista musi spełnić pewne warunki których nie omówiono w 8.2, a mianowicie:

- przy definiowaniu okręgu w postaci kanonicznej, podawany w definicji wektor (tzn. czwarty, piąty i szósty parametr w postaci kanonicznej) powinien być wektorem $(0,0,1)$,
- wektor prostopadły podawany w definicji prostej (LINE) lub płaszczyzny (PLANE) powinien być wektorem jednostkowym, podanym z dokładnością przynajmniej do sześciu cyfr,
- wektor definiujący kierunek osi walca (CYLINDR) lub stożka (CONE) powinien być wektorem jednostkowym, określonym poprawnie przynajmniej do sześciu cyfr.

33.2. Interpretacja wyników przy definiowaniu walca tabelarycznego

Zdefiniowany walec tabelaryczny (TABCYL) składa się z serii gładko ze sobą połączonych krzywych. Utworzona w ten sposób pojedyncza krzywa przedstawiona jest za pomocą całego zbioru punktów wejściowych. Każda pojedyncza krzywa lub odcinek rozciąga się od jednego punktu wejściowego do następnego, w kolejności pojawiania się punktów na wejściu, krzywą tę lub odcinek przedstawia równanie przestrzenne. Równanie obowiązuje w lokalnym układzie współrzędnych, którego początkiem jest pierwszy wprowadzony punkt, zaś oś zmiennej niezależnej jest linią przechodzącą przez dwa końcowe punkty pierwszego odcinka. Dodatni kierunek osi zmiennej niezależnej ustalony jest od początku odcinka w kierunku drugiego punktu. Równanie przestrzenne dla każdego odcinka może być zapisane w postaci:

$$Q = AR^3 + BR^2 + CR$$

Współczynniki A i B są zapamiętane w postaci kanonicznej walca tabelarycznego (ich położenie podane będzie później), zaś C może być obliczone jako:

$$C = -(AL^2 + BL)$$

gdzie L jest długością odcinka (odległością między dwoma punktami krańcowymi), która jest również zapamiętana w postaci kanonicznej (zob. pkt 8.1). Punkty na początku podane w celu zdefiniowania walca tabelarycznego są następnie przekształcone do układu współrzędnych UVW takiego, że wszystkie punkty tego walca leżą w płaszczyźnie UV (czyli $W=0$). Transformacja następuje tylko przez obrót - początek układu współrzędnych UVW leży w tym samym miejscu, co początek układu współrzędnych XYZ, w którym został zdefiniowany walec tabelaryczny. Przekształcenie powyższe nie ma żadnego związku z przekształceniem określonym przez modyfikator TRFORM w definicji walca tabelarycznego.

Macierz o wymiarach 3×3 , opisująca przekształcenie z układu współrzędnych XYZ do układu współrzędnych UVW jest zapamiętana na początku postaci kanonicznej walca tabelarycznego.

Należy zauważyć, że macierz ta jest jednostkowa, gdy obrót jest zerowy.

Po tym przekształceniu punktów wyjściowych, wywoływane jest równanie definiujące krzywą, składającą się z pojedynczych odcinków i dane definiujące każdy odcinek są dołączone do postaci kanonicznej tabelarycznego walca. Dla każdego odcinka zostaje zapamiętanych siedem części informacyjnych. Są to w kolejności:

- współrzędne U i V punktu początkowego odcinka,
- współczynniki A i B równania przestrzennego oraz odcinka (o czym była mowa poprzednio),
- maksymalna i minimalna wartość sześciianu (wartość Q w równaniu).

Maksymalna i minimalna wartość są wyrażone w postaci zwiolokrotnionej długości odcinków, do których się odnoszą np. minimalna wartość 0.5 dla odcinka określa wartość Q , która równa się iloczynowi 0.5 przez długość odcinka, zaś maksymalna wartość 3.1 określa wartość Q , która równa się iloczynowi 3.1 przez długość odcinka, do którego się to maksimum odnosi.

Należy również zaznaczyć, że system APT dodaje jeden własny punkt (i w konsekwencji jeden odcinek) na końcu definicji walca tabelarycznego. Każdy z tych końcowych punktów jest zlokalizowany poza sąsiadującym z nim punktem końcowym określonym przez programistę, na linii prostej o takim samym nachyleniu, jak ostatni podany przez programistę odcinek.

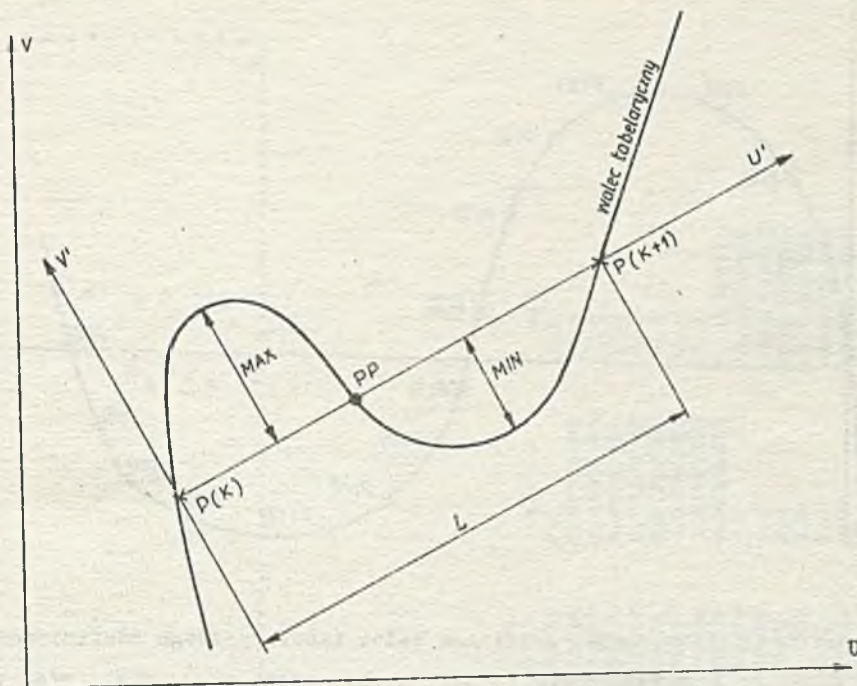
W chwili gdy jest definiowany walec tabelaryczny, zwykle odcinki mają długość 10 jednostek, ale odległość ta może być skrócona, jeżeli walec tabelaryczny owija się w ten sposób, że "rozszerzające odcinki" przecinałyby się. W momencie gdy walec tabelaryczny jest definiowany, drukowane są pewne dotyczące go informacje tak, że programista może sprawdzić odpowiedniość krzywej, którą stworzył system na podstawie jego punktów. Jeżeli programista poda modyfikator SPLINE dla walca tabelarycznego, to pierwszą wydrukowaną informacją jest listing i rysunek krzywej walca tabelarycznego podany w każdym punkcie wyjściowym. Rysunek jest dosyć niedokładny i pokazuje tylko charakter zmian krzywizny wzdłuż walca. Określona w każdym punkcie wartość krzywizny (do szóstciu miejsc dziesiętnych) jest drukowana po lewej stronie rysunku i każda wartość jest drukowana w tym samym wierszu, w którym na rysunku pojawia się punkt właśnie o takiej krzywiznie. Wartość krzywizny poprzedzona jest przez kolejny numer danego punktu wejściowego, a więc punkt dodatkowy nie jest tu wliczany.

Informacje drukowane pod rysunkiem krzywizny przy podaniu modyfikatora SPLINE dla walca tabelarycznego są takie same jak te, które są drukowane dla walców skonstruowanych za pomocą innych dostępnych metod. Składają się na nie: długość walca tabelarycznego w postaci kanonicznej, po niej następuje macierz transformacji z układu XYZ do UVW i liczba punktów pokrywających walec (wliczając w to dwa dodatkowe punkty). Na końcu drukowana jest pozostała część postaci kanonicznej walca tabelarycznego, w siedmiu kolumnach odpowiadających siedmiu rodzajom danych opisujących każdy zdefiniowany odcinek. Dla każdego odcinka w skrajnie lewej kolumnie podawane są współrzędne u , v punktu początkowego.

Na rys. 244 przedstawiony jest wyjściowy listing definicji walca tabelarycznego pokazanego na rys. 243. Dla wygody współrzędna u jest poprzedzona przez numer punktu, do którego się odnosi, ale numery punktów nie pojawiają się przy zwykłym listingu z maszyny cyfrowej.

Ponizej zasygnalizowane będą pewne problemy, które należy wziąć pod uwagę przy analizie wyników walca tabelarycznego.

- Niezerowa wartość maksymalna i minimalna dla tego samego odcinka wskazują na punkt przecięcia (zob. pkt PP na rys. 242). Zwykle takie odwrócenie krzywizny nie jest pożądane. Przez tego



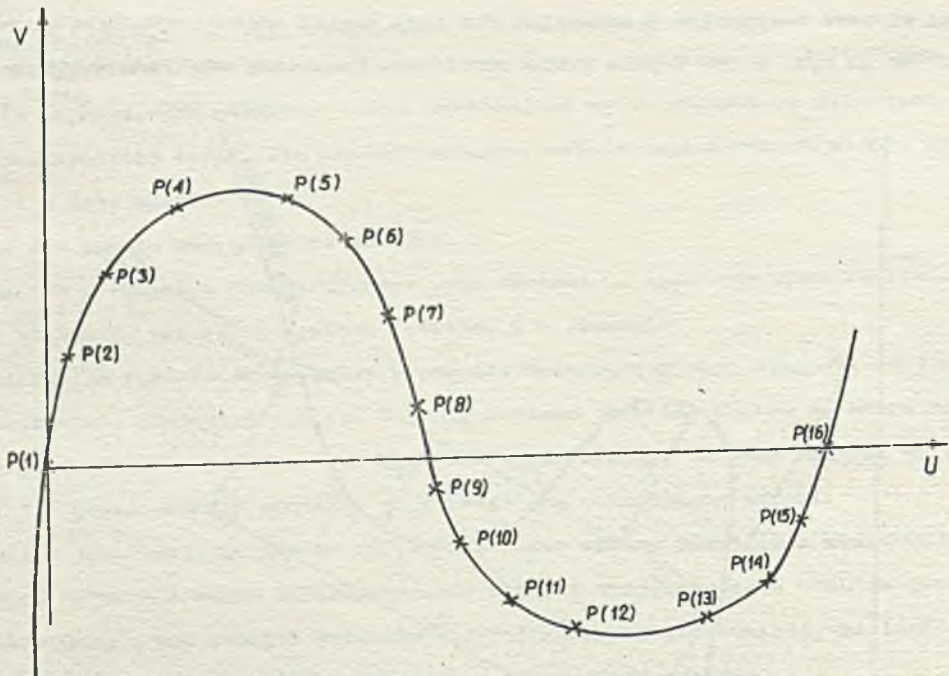
Rys. 242. Segment walca tabelarycznego

rodzaju zmiany (odwrócenie krzywizny) lub zastosowanie wzoru zmian wartości zerowych między maksymalną a minimalną kolumną przez kilka odcinków, można uzyskać bardzo nierówną krzywą. Może to wynikać ze zbyt dużej lub ze zbyt małej liczby punktów wejściowych.

- Maksymalna i minimalna wartość powinny być dokładnie zbadane pod kątem odchylenia od średniej. Dowlone wartości, które wskazują znaczne przekroczenie wartości przeciętnej, określają wierzchołki walca tabelarycznego. Aktualna wysokość (lub głębokość) może być obliczona przez pomnożenie interesującej wartości przez długość podaną w kolumnie 5.
- Szybkie zmiany krzywizny na rysunku, jeśli występują, mogą też wskazywać na ubogie dane wejściowe lub dane przyległe do pola zmian. Zależą one od skali rysunku drukowanego w postaci zbioru punktów. Pojedynczy punkt nieciągłości niekoniecznie musi wskazywać na pewną trudności (zob. wykres na rys. 244 dla pola punktów 8 i 9 walca tabelarycznego, którego kształt jest przedstawiony na rys. 243).

33.3. Instrukcja DEBUG

Instrukcja DEBUG powoduje wydruk wyników pośrednich operacji przeprowadzanych w fazie translacji programu obróbki części. Jeżeli podczas późniejszego przetwarzania programu, w drugiej fazie pracy systemu APT pojawi się błąd, wówczas będą wyprowadzone wszystkie zmienne wspólne.



Rys. 243. Ilustracja graficzna walca tabelarycznego zdefiniowanego instrukcją:

TAB=TABCYL/NOZ, P (1), P (2), P(3), P(4), P(5), P(6), P(7), \$
P(8), P(9), P(10), P(11), P(12), P(13), P(14), P(15), P(16)

Instrukcja DEBUG ma następującą postać:

$$\text{DEBUG/} \left[\begin{array}{c} \text{SEC1} \\ \text{SEC2} \end{array} \right], \left[\begin{array}{c} \text{ON} \\ \text{OFF} \end{array} \right]$$

gdzie:

SEC1 - oznacza, że system ma pracować w trybie diagnostycznym w fazie translacji,

SEC2 - oznacza, że system ma pracować w trybie diagnostycznym w fazie pracy jednostki arytmetycznej,

ON - jest modyfikatorem oznaczającym rozpoczęcie pracy w trybie diagnostycznym,

OFF - jest modyfikatorem oznaczającym zakończenie pracy w trybie diagnostycznym.

Instrukcja DEBUG powoduje wyprowadzanie wyników pośrednich z odpowiedniej fazy pracy systemu APT.

Oznacza to, że jeżeli w instrukcji DEBUG zostanie podany modyfikator SEC1, wówczas będą wyprowa-

dzane informacje z fazy translacji, natomiast jeżeli zostanie podany modyfikator SEC2, wówczas

będą wyprowadzane informacje z fazy pracy jednostki arytmetycznej. Instrukcja DEBUG powoduje wy-

prowadzenie wszystkich błędów oraz związanych z nimi numerów instrukcji, jeśli błędy pojawią s-

w programie obróbki części. Dodatkowo drukowane są informacje wyjściowe DYNDMP, opisane w punkto-

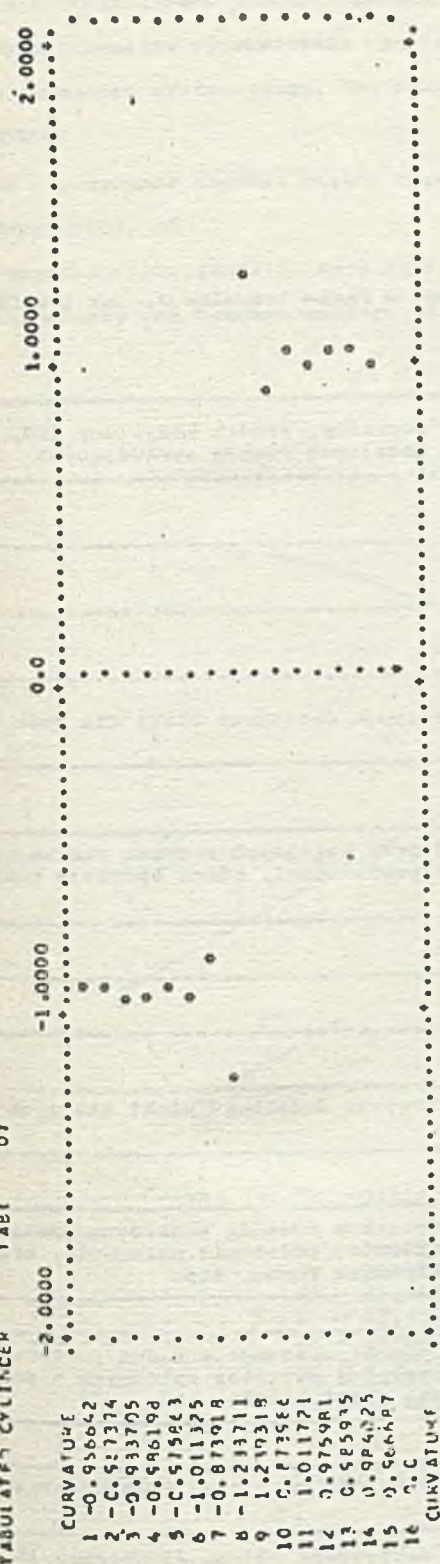
33.4. Jeżeli w programie obróbki części pojawi się instrukcja w postaci

$$\text{DEBUG/} \left[\begin{array}{c} \text{ON} \\ \text{OFF} \end{array} \right]$$

wówczas system APT pracuje w trybie diagnostycznym dla grupy wybranych instrukcji, zarówno w fa-

zie translacji, jak i w fazie pracy jednostki arytmetycznej. Grupa instrukcji, dla których

77
C(1),C(2),C(3),C(4),C(5),C(6),C(7),C(8),C(9),C(10),C(11),C(12),C(13),C(14),C(15),C(16)
 TABULATED CYLINDER TAB 01



DATA STORAGE = 132

ROTATION MATRIX

1.0000000 0.0 0.0
 C.C 1.0000000 0.0
 0.0 0.0 1.0000000

NUMBER OF POINTS = 14

1 -4.70746474
 2 3.12500000
 3 0.37500000
 4 0.62500000
 5 0.87500000
 6 1.12500000
 7 1.37500000
 8 1.62500000
 9 1.87500000
 10 2.12500000
 11 2.37500000
 12 2.62500000
 13 2.87500000
 14 3.12500000
 15 3.37500000
 16 3.62500000
 17 3.87500000
 18 4.12500000
 FINI

LENGTH MAX MIN
 1 0.0000000 0.0 0.0
 2 0.38783158 0.04966290 0.0
 3 0.28971189 0.03646269 0.0
 4 0.25834506 0.03260814 0.0
 5 0.25000000 0.03139213 C.C
 6 0.25834506 0.03292182 0.0
 7 0.28971189 0.03518172 0.0
 8 0.38783158 0.05511959 0.0
 9 0.38783158 -0.02136137 -0.0213715
 10 0.28971189 -0.05514932 -0.05516379
 11 0.25834506 -0.01293697 -0.01293697
 12 0.25000000 -0.01383336 -0.01383336
 13 0.25834506 -0.03261040 -0.03261040
 14 0.28971189 -0.03646269 -0.03646269
 15 0.38783158 -0.04966290 -0.04966290
 16 0.00000000 0.0 0.0

Rys. 244. Informacje drukowane o walcu tabelarycznym

system APT pracuje w trybie diagnostycznym zawiera się między parą instrukcji DEBUG/ON oraz DEBUG/OFF. Należy podkreślić, że jeżeli w programie obróbki części pojawiają się obie postaci instrukcji DEBUG tzn.

DEBUG/SEC1, $\begin{bmatrix} \text{ON} \\ \text{OFF} \end{bmatrix}$ oraz DEBUG/SEC2 , $\begin{bmatrix} \text{ON} \\ \text{OFF} \end{bmatrix}$

jest to równoznaczne z pojawieniem się instrukcji

DEBUG/ $\begin{bmatrix} \text{ON} \\ \text{OFF} \end{bmatrix}$

a więc system APT pracuje w trybie diagnostycznym, zarówno w fazie translacji, jak i w fazie pracy jednostki arytmetycznej.

Dane o powierzchni: punkt leżący na powierzchni, wektor normalny, środek krzywizny itd. generowane dla każdej powierzchni podczas przetwarzania kolejnych ruchów skrawających

Pierwszy blok Powierzchnia przedmiotu

Drugi blok Powierzchnia prowadząca

Trzeci blok Powierzchnia ograniczająca

(w wypadku wielokrotnych powierzchni ograniczających występują dodatkowe bloki dla tych powierzchni)

Etykiety i przełączniki do każdej powierzchni potrzebnej przy kolejnych ruchach skrawających, wskazujące na ustawienie narzędzia skrawającego względem powierzchni, różne operacje realizowane przez jednostkę arytmetyczną, itp.

Czwarty blok Powierzchnia przedmiotu

Piąty blok Powierzchnia prowadząca

Szósty blok Powierzchnia ograniczająca

(w wypadku wielokrotnych powierzchni ograniczających, występują dodatkowe bloki dla tych powierzchni)

Siódmy blok Obliczone dane, które nie są powiązane z żadną konkretną powierzchnią. Zawiera takie informacje, jak aktualne położenie narzędzia, orientacja osi narzędzia, aktualny kierunek ruchu, itp.

Ósmy blok Etykiety i przełączniki, które nie są związane z żadną powierzchnią. Zawiera takie informacje, jak powrotne wartości zmiennych z podprogramów ARELEM, które są aktualnie przetwarzane, itd.

Dziewiąty blok Rekordy PROTAP ozytane przez ARELEM dla kolejnych ruchów skrawających.

Dziesiąty blok Postacie kanoniczne używanych powierzchni. Powierzchnie zapamiętane są w kolejności PS, DS, CS, bez spacji między zapisami.

Rys.245. Format danych wyjściowych uzyskanych za pomocą instrukcji DYNDMP

33.4. Instrukcja DYNDMP

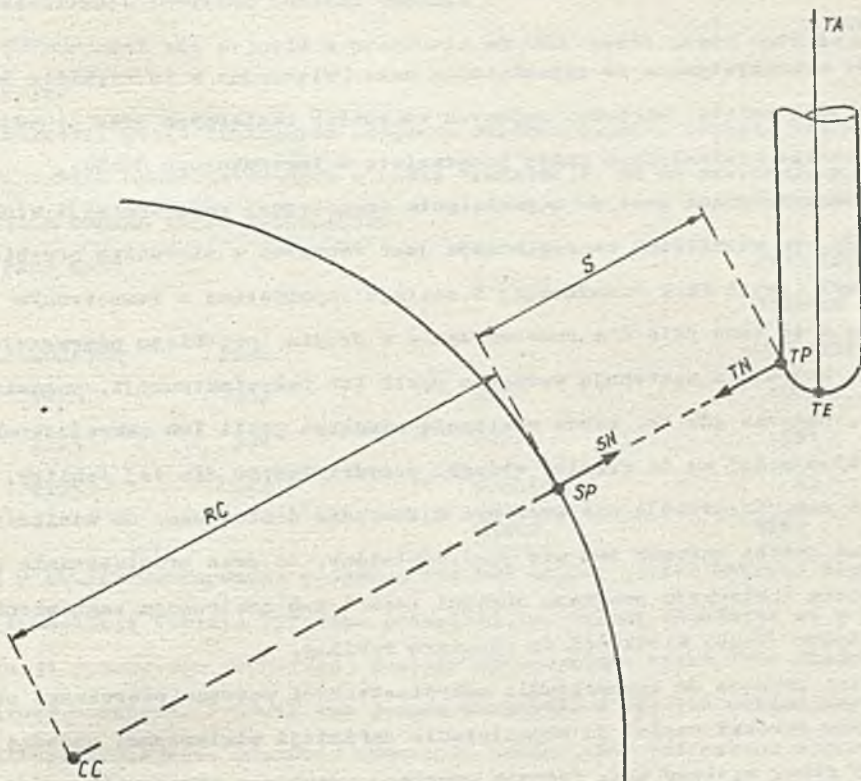
Instrukcja DYNDMP pozwala programistom systemowym na dynamiczne składowanie programu obróbki części. Ponadto wyprowadzane wyniki umożliwiają dodatkowo śledzenie obliczeń wykonywanych przez jednostkę arytmetyczną. Instrukcja ta ma postać:

DYNDMP

Jeżeli w programie obróbki części oprócz instrukcji DYNDMP wystąpi instrukcja

DEBUG/SEC2, ON

to w wypadku wystąpienia błędu w fazie pracy jednostki arytmetycznej będą wyprowadzone odpowiednie komunikaty dla każdego takiego błędu.



Rys. 246 Oznaczenia stosowane w podręczniku języka APT:

- CC - środek krzywizny
- RC - promień krzywizny
- SP - punkt powierzchni
- SN - prostopadła do powierzchni
- S - odległość
- TP - punkt narzędzia
- TN - prostopadła do narzędzia
- TE - punkt końcowy narzędzia
- TA - oś narzędzia

Ze względu na to, że stosowanie instrukcji DYNDMP wymaga pewnych dodatkowych wiadomości o działaniu systemu APT - wykraczających poza zakres tego podręcznika - działanie jej będzie omówione bardzo pobieżnie. Na rys. 245 został przedstawiony format informacji wyprowadzanych przy pomocy instrukcji DYNDMP, natomiast na rys. 246 - znaczenie symboli używanych przez

instrukcję DYNDMP.

Zmienna IS podana w ósmym bloku danych (rys. 245) określa, którą powierzchnię w danym momencie zajmuje się system APT, a mianowicie gdy:

- IS=1 - powierzchnią przedmiotu,
- IS=101 - powierzchnią prowadzącą,
- IS=201 - powierzchnią ograniczającą,
- IS=301 - drugą powierzchnią ograniczającą.

33.5. Instrukcja TUNEUP

W fazie translacji programu obróbki części system APT tworzy trzy tablice podstawowe: VST, PTPP i CANON.

Tablica VST jest wykorzystywana do zapamiętania nazw (włączając w to etykiety instrukcji) definiowanych przez programistę, wartości nadanych zmiennych skalarnych oraz pozycji dla poszczególnych wskaźników tablic wymienionych przez programistę w instrukcjach RESERV.

Tablica PTPP wykorzystywana jest do zapamiętania wewnętrznej reprezentacji większości instrukcji języka APT. Ta wewnętrzna reprezentacja jest tworzona w pierwszym przebiegu pierwszej fazy pracy systemu APT (czyli fazy translacji) i zostaje zapamiętana w zewnętrznym zbiorze, skąd jest później czytana w to samo pole dla przetwarzania w drugim przebiegu pierwszej fazy pracy systemu. Instrukcje, które nie występują wewnątrz pętli lub makroinstrukcji, zapamiętywane są w tablicy tylko raz, podczas gdy te, które występują wewnątrz pętli lub makroinstrukcji zapamiętywane są razem, w kolejności aż do granicy obszaru przydzielonego dla tej tablicy. Pojedyncza pętla lub makroinstrukcja nie musi być całkowicie dostosowana do wielkości tablicy za jednym razem, chociaż jeżeli warunek ten nie jest spełniony, to czas przetwarzania przedłuża się. Jednak każda pojedyncza instrukcja programu obróbki części lub instrukcja zagnieżdżona musi być dostosowana pod względem swojej wielkości do rozmiaru tablicy.

Tablica CANON jest używana do zapamiętania makroinstrukcji podczas pierwszego przebiegu przy przetwarzaniu programu obróbki części do zapamiętania definicji powierzchni podczas drugiego przebiegu pierwszej fazy pracy systemu APT. Podczas drugiego przebiegu wykorzystywane jest też do pewnych funkcji wewnętrznych pole CANON o wielkości 1250 słów.

Przydzielony dla tych tagów obszar pamięci podczas pierwszego przebiegu pierwszej fazy pracy systemu APT zwykle wynosi:

2750 słów dla tablicy VST, po 2225 słów dla tablic PTPP i CANON. Pod koniec pierwszego przebiegu znane są całkowite wymagania w stosunku do tablic VST i PTPP. Nie wykorzystany obszar tych tablic jest automatycznie dołączany do obszaru tablicy CANON do wykorzystania w drugim przebiegu. Jeżeli obszar pamięci żądany przez jedną z tych tablic dla programu obróbki części okazuje się być większy niż zwykle przydzielany, to rozmiary obszarów mogą być zmieniane za pomocą instrukcji TUNEUP. Nie używany obszar może być "pożyczony" z jednej tablicy w celu rozszerzenia pojemności innej. Muszą być przy tym spełnione pewne warunki, a mianowicie całkowity obszar zajmowany przez te trzy tablice nie może przekraczać 7200 słów, zaś obszar przeznaczony dla tablicy CANON nie może być mniejszy niż 1300 słów (aby móc wykorzystać 1250 słów dla celów specjalnych i minimum

50 słów pamięci dodatkowej).

Instrukcja TUNEUP ma następującą postać:

TUNEUP / i, j, k

gdzie:

- i - jest rozmiarem pamięci zarezerwowanej dla tablicy VST podczas pierwszego przebiegu przetwarzania programu obróbki części,
- j - jest rozmiarem pamięci zarezerwowanej dla tablicy PTPP podczas pierwszego przebiegu przetwarzania programu obróbki części,
- k - jest rozmiarem pamięci zarezerwowanej dla tablicy CANON podczas pierwszego przebiegu przetwarzania programu obróbki części.

Instrukcja TUNEUP musi się pojawić w programie obróbki części przed definicją dowolnej pętli lub makroinstrukcji.

W fazie translacji, przed listingiem programu obróbki części, pojawia się wydruk informacyjny o rozmiarach tablic wykorzystywanych w fazie translacji. Ma on następującą postać

TABLE USAGE DURING INPUT TRANSLATION

PASS ONE			PASS TWO	
			DYNAMIC	
	ALLOCATED	USED	ALLOCATION	
VST	2750	1902	VST	1902
PTPP	2225	394	PTPP	394
CANON	2225	155	SCALARS	23
			CANON	3881

Załóżmy, że w czasie wykonywania programu obróbki części został wykryty błąd nr 15. Oznacza to, że w fazie translacji tablica VST jest przepełniona. Należy zauważyć, że w początkowej części fazy translacji (pierwszym przebiegu) zostało wykorzystane tylko 1902 miejsca tablicy VST spośród 2750 zarezerwowanych. Pojawił się jednak komunikat o błędzie wskazujący, że liczba miejsc 2750 jest niewystarczająca, ponieważ utworzona tablica VST potrzebuje więcej niż 2750 pozycji.

Ponieważ nie był dostępny wystarczający obszar pamięci, to dalsze odwołania do tablicy VST dla tego fragmentu programu obróbki części zostały pominięte przy późniejszym przetwarzaniu. Liczba 1902 wskazuje, że wymagane jest rozszerzenie obszaru pamięci przydzielonego dla tablicy VST, aby można było wykonać cały program obróbki części.

Trzeba więc podać instrukcję TUNEUP, która przydzieli odpowiedni obszar pamięci tablicy VST. Obszar pamięci zarezerwowany dla tablicy CANON 3881 jest zbyt rozległy. Podobnie dla tablicy PTPP gdzie wykorzystanych jest tylko 394 miejsca. A więc brakujący do rozszerzenia tablicy VST obszar pamięci może być pożyczony z tablic CANON oraz PTPP.

Odpowiednia instrukcja TUNEUP ma następującą postać:

TUNEUP/3500, 1000, 2700

Instrukcja ta przydziela tablicy VST 3500 miejsce, tablicy PTPP - 1000 miejsce, zaś tablicy CANON - 2700 miejsce, co w sumie daje 7200 miejsce.

Poniżej przedstawiono wydruk z kolejnego wykonania programu obróbki części, z wykorzystaniem powyższej instrukcji TUNEUP.

TABLE USAGE DURING INPUT TRANSLATION

PASS ONE			PASS TWO	
	ALLOCATED	USED	DYNAMIC	
			ALLOCATION	
VST	3500	3339	VST	3339
PTPP	1000	394	PTPP	394
CANON	2700	155	SCALARS	23
			CANON	2444

Wykonanie programu obróbki części zostało teraz prawidłowo zakończzone. Należy zauważyć, że potrzebny obszar dla tablicy VST wynosił 3339 miejsce, a więc przekroczył on znacznie zwykle przydzielony obszar 2750 miejsce.

Programy obróbki części z wielu przyczyn mogą powodować przekroczenie przyznanego limitu dla tablic VST i PTPP. W większości wypadków łączenie obszarów przeznaczonych na wszystkie tablice za pomocą instrukcji TUNEUP pozwala na ukończenie przetwarzania programów obróbki części. W sytuacjach, gdy instrukcja TUNEUP nie może poprawić przetwarzania programu obróbki części, programista może spróbować zmniejszyć liczbę pozycji w tablicy VST przez zmianę definicji geometrycznych:

- użycie instrukcji CANON/ON,
- użycie słowa CANON jako części definicji geometrycznej i następnie ponowne użycie tej samej nazwy,
- zagnieżdżenie nienazwanych definicji geometrycznych w instrukcjach.

Przeddefiniowania geometryczne mogą być potrzebne, jeżeli instrukcja RESERV zawiera duże wartości lub dużą liczbę nazw. Sumując liczby wskaźników podanych w instrukcji RESERV, możemy określić potrzebną liczbę słów. Ilustruje to poniższy przykład.

Przykład 1

Poniższa instrukcja:

RESERV/A,500,X,3000,Y,5

powoduje zarezerwowanie 3511 słów w tablicy VST, co wynika ze zsumowania długości tablic A,X,Y oraz dodatkowo po 2 słowa na każdą nazwę podaną w instrukcji RESERV.

Jeżeli program obróbki części używający dużą liczbę zmiennych indeksowanych wykorzysta całe miejsca w tablicy i przeddefiniowania nie są możliwe, to wymagana jest segmentacja programu obróbki części.

Inną sytuację, powodującą przepełnienie tablicy VST jest użycie etykiet instrukcji wewnątrz makroinstrukcji. Za każdym razem, gdy wywoływana jest nazwa makroinstrukcji, to zbiór etykietowanych instrukcji jest zapamiętany w tablicy VST. Jeżeli program obróbki części często woła makroinstrukcje, to wiele zbiorów etykietowanych instrukcji będzie zapamiętanych w tablicy VST, co może spowodować jej szybkie przepełnienie. Ilustruje to poniższy przykład.

Przykład 2

W makroinstrukcji jest 10 etykietowanych instrukcji, zaś w programie obróbki części występuje

10 odwołań. W sytuacji tej zostanie wykorzystanych:

$10 \times 2 \text{ słowa} \times (1 + 10 \text{ odwołań})$, czyli

220 słów w tablicy VST.

Jeżeli program obróbki części zawiera wiele makroinstrukcji z wewnętrznymi etykietowanymi instrukcjami, to problem ten można jedynie rozwiązać przez "pożyczenie" miejsca z innych tablic za pomocą instrukcji TUNEUP lub segmentację programu obróbki części.

D O D A T E K A

Spis słów kluczowych języka APT

Poniżej będzie podany zbiór słów kluczowych języka APT (zarówno słów głównych jak i modyfikatorów). Ze względu na to, że modyfikatory są zwykle używane w różnych instrukcjach języka APT i w związku z tym są opisane w różnych częściach podręcznika, to przy nazwie takiego modyfikatora (który jest stosowany wielokrotnie) będzie podana jedynie odpowiednia informacja, bez podawania numerów rozdziałów oraz stron. Przy słowach kluczowych głównych oraz modyfikatorach, które są używane tylko w jednej instrukcji, będzie podany numer rozdziału oraz numer strony w odpowiedniej części podręcznika, na której słowo to jest opisane

AAXIS	modyfikator		
ABSF	4.2	I 16	
ADJUST	modyfikator		
AIR	32.2	III	67
ALL	modyfikator		
ANGLF	4.2	I 17	
ARC	5.2	I 28	
ARCSLP	32.2	III	67
ASLOPE	32.2	III	67
AT	5.2	I 27, 30, 31	
ATANF	4.2	I 16	
ATAN2F	4.2	I 16	
ATANGL	modyfikator		
AUTO	modyfikator		
AUTOPS	14.1	II 19	
AUXFUN	32.2	III	67
AVOID	13.4	II 16	
BAXIS	modyfikator		
BCD	modyfikator		
BENCD	30.5	III	56
BENDIX	30.5	III	56
BINARY	modyfikator		
BLACK	modyfikator		

BLUE	modyfikator		
BORE	modyfikator		
BOTH	modyfikator		
BURG	30.5	III	56
CALL	24.3	III	29
CAMERA	32.2	III	67
CANON	8.2	I 101	
CAXIS	modyfikator		
CCLW	5.2	I 28	
CENTER	modyfikator		
CHECK	32.2	III	67
CHUCK	modyfikator		
CINCY	30.5	III	56
CIRCLE	5.5	I 48	
CIRCUL	modyfikator		
CIRLIN	modyfikator		
CLAMP	32.2	III	67
CLEARP	32.2	III	67
CLPRNT	30.3	III	
CLRSRF	32.2	III	67
CTLV	30.1	III	53
CLW	5.2	I 28	
COLLET	modyfikator		
CONCRD	30.5	III	56
CONE	5.13	I 67	
CONSEC	5.18	III dod E	
CONST	13.4	II 16	
COOLNT	32.2	III	67
COPY	19.2, 32.2	II 79, III	68
CORNF	32.2	III	68
COSF	4.2	I 16	
COUPLE	32.2	III	68
CROSS	modyfikator		
CUT	31.4	III	60
CUTANG	modyfikator		
CUTCOM	32.2	III	68
CUTTER	11	II 5	
CYCLE	32.2	III	68
CYLNR	5.12	I 67	
DARK	modyfikator		
DASH	modyfikator		

DEBUG	33.3	III	75
DECR	modyfikator		
DEEP	modyfikator		
DELAY	32.2	III	68
DEDETE	32.2	III	68
DISPLY	32.2	III	68
DISTF	4.2	I 16	
DITTO	modyfikator		
DNTCUT	31.4	III	60
DNTCUT/PRINT	31.4	III	60
DOTF	4.2	I 16	
DOTTED	modyfikator		
DOWN	modyfikator		
DRAFT	32.2	III	68
DRAWLI	32.1	III	66
DRESS	32.1	III	66
DRILL	modyfikator		
DSTAN	modyfikator		
DVLIEG	30.5	III	56
DWELL	32.2	III	68
DYNDMP	33.4	III	79
DYNPAT	30.5	III	56
ECS	30.5	III	56
ELLIPS	5.6	I 55	
END	32.1	III	66
ENDARC	modyfikator		
EXPF	4.2	I 16	
EXPOST	30.5	III	56
FACE	modyfikator		
FACEML	32.1	III	66
FEDRAT	32.2	III	68
FINI	20.3	II 87	
FLOOD	modyfikator		
FOSDIK	30.5	III	56
FOURPT	modyfikator		
FROM	13.1	II 9	
FRONT	modyfikator		
FULL	modyfikator		
FUNOFY	modyfikator		
GAPLES	modyfikator		
GCONIC	5.8	I 57	

GECENT	30.5	III	56
GO	15.2	II 26	
GOBACK	16.1	II 47	
GOCLER	32.1	III	66
GODLTA	13.3	II 11	
GODOWN	16.1	II 47	
GOFWD	16.1	II 47	
GOHOME	32.1	III	66
GOLFT	16.1	II 47	
GORGT	16.1	II 47	
GOTO	13.2	II 11	
GOUGCK	31.6	III	64
GOUP	16.1	II 47	
GREEN	modyfikator		
HEAD	32.2	III	68
HIGH	modyfikator		
HOLDER	modyfikator		
HYPERB	5.7	I 57	
IF	23.3	III	25
IFRO	32.2	III	68
IN	modyfikator		
INCR	5.2	I 27	
INDEX	19.2, 32.2	II 79, III	68
INDIRP	15.1	II 24	
INDIRV	15.1	II 24	
INSERT	26.4, 32.2	III	47, III 68
INTCOD	32.2	III	68
INTENS	modyfikator		
INTERC	modyfikator		
INTOF	modyfikator		
INTOL	12	II 7	
INVERS	13.4	II 13	
IPM	modyfikator		
IPR	modyfikator		
ISTOP	32.1	III	66
JUMPTO	23.4	III	26
LARGE	modyfikator		
LAST	modyfikator		
LCONIC	5.9	I 58	
LEADER	32.2	III	68
LEFT	modyfikator		

LENGTH	modyfikator		
LETTER	32.2	III	68
LIBRY	24.7	III	39
LIGHT	modyfikator		
LINE	5.4	I 37	
LINCIR	modyfikator		
LINEAR	5.2	I 26	
LITE	modyfikator		
LNTHF	4.2	I 16	
LOADTL	32.2	III	68
LOCK	modyfikator		
LOCKX	32.1	III	66
LOFT	5.18	III dod. E	
LOGF	4.2	I 16	
LOG10F	4.2	I 16	
LOOPND	23.2	III	25
LOOPST	23.1	III	25
LOW	modyfikator		
LPRINT	32.2	III	68
MACHIN	30.5, 32.2	III 56, III	68
MACRO	24.1	III	29
MAGTAP	32.2	III	69
MAIN	modyfikator		
MANUAL	modyfikator		
MATRIX	6.2	I 84	
MAXDP	31.3	III	59
MAXDPM	modyfikator		
MAXIPM	modyfikator		
MAXRPM	modyfikator		
MAXVEL	32.2	III	69
MCHFIN	32.2	III	69
MCITOL	32.2	III	69
MDEND	32.2	III	69
MDWRIT	32.2	III	69
MED	modyfikator		
MEDIUM	modyfikator		
MESH	5.18	III dot. E	
MILL	modyfikator		
MILWAK	30.5	III	56
MINUS	modyfikator		
MIRROR	modyfikator		

MIST	modyfikator		
MODE	32.2	III	69
MODIFY	modyfikator		
MOVETO	32.2	III	69
MULTAX	22.1	III	5
MULTRD	modyfikator		
NDTEST	31.1	III	57
NEGX	modyfikator		
NEGY	modyfikator		
NEGZ	modyfikator		
NEXT	modyfikator		
NIXIE	modyfikator		
NOMORE	modyfikator		
NOPLOT	30.4	III	56
NOPOST	30.6	III	57
NOPS	15.2	II 29	
NORMAL	modyfikator		
NORMDS	modyfikator		
NORMPS	modyfikator		
NOW	modyfikator		
NOX	modyfikator		
NOY	modyfikator		
NOZ	modyfikator		
NUMF	4.2	I 17	
NUMPTS	31.2	III	58
OBTAIN	27	III	49
OFF	modyfikator		
OFFSET	15.4	II 42	
OMIT	13.4	II 14	
ON	modyfikator		
OPEN	modyfikator		
OPSKIP	32.2	III	69
OPSTOP	32.1	III	66
OPTION	modyfikator		
ORIGIN	32.2	III	69
OUT	modyfikator		
OUTTOL	12	II 7	
OVPLOT	32.2	III	69
PARAB	modyfikator		
PARLEL	modyfikator		
PART	modyfikator		

PARTNO	20.1, 32.2	II 87, III	69
PAST	modyfikator		
PATERN	5.2	I 26	
PBS	32.2	III	69
PEN	modyfikator		
PENDWN	32.1	III	66
PENUP	32.1	III	66
PERPTO	modyfikator		
PERSP	modyfikator		
PICKUP	32.1	III	66
PITCH	32.2	III	69
PIVOTZ	32.2	III	69
PLABEL	32.2	III	69
PLANE	5.10	I 60	
PLOT	32.2	III	69
PLUNGE	32.2	III	69
PLUS	modyfikator		
POCKET	18	II 69	
POINT	5.1	I 19	
POLCON	5.16	I 74	
POSTN	32.2	III	69
POSX	modyfikator		
POSY	modyfikator		
POSZ	modyfikator		
PLOT	32.2	III	69
PRINT	26.4, 32.2	III 48,	69
PRATTW	30.5	III	56
PREFUN	32.2	III	69
PRINT	26.1	III	42
PROBOG	30.5	III	56
PROBX	32.1	III	66
PRODY	32.1	III	66
PSIS	14.1	II 19	
PSTAN	modyfikator		
PTNORM	modyfikator		
PTONLY	30.2	III	54
PISLOP	modyfikator		
PTW4E	30.5	III	56
PUNCH	26.2	III	45
QUADRIC	5.14	I 68	
RADIUS	modyfikator		

RAIL	modyfikator		
RANDOM	5.2	I 26	
RANGE	modyfikator		
RAPID	32.1	III	66
READ	26.3	III	46
REAM	modyfikator		
REAR	modyfikator		
RED	modyfikator		
REFSYS	6.1	I 81	
REGBRK	32.2	III	69
REMARK	20.2	II 87	
RESERV	3.1	I 12	
RESET	32.1	III	66
RETAIN	13.4	II 14	
RETRCT	32.1	III	66
REV	modyfikator		
REVERS	32.2	III	69
REWIND	32.1, 32.2	III 66, 70	
RIGHT	modyfikator		
RLDSRF	5.17	I 76	
ROTABL	32.2	III	70
ROTHED	32.2	III	70
ROTREF	modyfikator		
RPM	modyfikator		
RTHETA	modyfikator		
SADDLE	modyfikator		
SAFETY	32.2	III	70
SAME	modyfikator		
SC4ø2ø	30.5	III	56
SCALE	modyfikator		
SCRIBE	modyfikator		
SECZRO	30.5	III	56
SEC1	25	III	42
SEC2	25	III	42
SELCTL	32.2	III	70
SEQNO	32.2	III	70
SETANG	modyfikator		
SETOOL	modyfikator		
SFM	modyfikator		
SIDE	modyfikator		
SINF	4.2	I 16	

SLOPE	modyfikator		
SLOWDN	32.2	III	70
SMALL	modyfikator		
SOLID	modyfikator		
SPHERE	5.11	I 64	
SPINDL	33.2	III	70
SPLINE	modyfikator		
SQRTF	4.2	I 16	
SRFVCT	15.3	II	
START	modyfikator		
STEP	modyfikator		
STOP	32.1	III	66
SUNTRN	30.5	III	56
SWITCH	32.1	III	66
SYN	28	III	50
TABCYL	5.15	I 70	
TANF	4.2	I 16	
TANTO	modyfikator		
TAP	modyfikator		
TAPKUL	modyfikator		
TERMAC	24.2	III	29
THETAR	modyfikator		
THICK	17	II 67	
THREAD	32.2	III	70
THRU	modyfikator		
TIMES	modyfikator		
TITLES	26.4	III	47
TLAXIS	22.2	III	7
TLLEFT	14.2	II 20	
TLNDON	14.2	II 20	
TLOFPS	14.1	II 18	
TLOX	14.2	II 20	
TLOXPS	14.1	II 18	
TLRGT	14.2	II 20	
TMARK	32.2	III	70
TO	modyfikator		
TOLER	12	II 7	
TOOL	5.18	III dod. E	
TOOLNO	32.2	III	70
TPI	modyfikator		

TRACUT	19.1, 32.2	II 77, III	70
TRANS	32.2	III	70
TRANSL	modyfikator		
TRANTO	31.5	III	64
TRAV	modyfikator		
TRFORM	modyfikator		
TRUTRA	30.5	III	56
TRW	30.5	III	56
TUNEUP	33.5	III	80
TUIN	modyfikator		
TUNRET	32.2	III	70
TWOPT	modyfikator		
TYPE	modyfikator		
UI.OKX	32.1	III	66
UNIT	modyfikator		
UNLOAD	32.1	III	66
UP	modyfikator		
VECTOR	5.3	I 33	
VTLAXS	22.3, 32.2	III16,	70
WCORN	22.3, 32.2	III22,	70
XAXIS	modyfikator		
XCOORD	modyfikator		
XLARGE	modyfikator		
XSMALL	modyfikator		
XYPLAN	modyfikator		
XYROT	modyfikator		
XYVIEW	modyfikator		
XYZ	modyfikator		
YCOORD	modyfikator		
YLARGE	modyfikator		
YSMALL	modyfikator		
YZPLAN	modyfikator		
YZROT	modyfikator		
YZVIEW	modyfikator		
ZAXIS	modyfikator		
ZCOORD	modyfikator		
ZERO	modyfikator		
ZLARGE	modyfikator		
ZSMALL	modyfikator		
ZSURF	5	I 19	
ZXPLAN	modyfikator		

ZXR0T	modyfikator		
ZXVIEW	modyfikator		
ZDCALL	31.1	III	57
JCALL	31.1	III	57
3PT2SL	modyfikator		
4.T1SL	modyfikator		
8PT	modyfikator		

D O D A T E K B

Komunikaty o błędach wykrywanych w fazie translacji

Numer błędu	Interpretacja
1.	Nazwa zmiennej składa się z więcej niż 6 znaków.
2.	Liczba składa się z więcej niż 17 cyfr.
3.	Uprzednio zdefiniowana nazwa została zdefiniowana jako etykieta instrukcji.
4.	Nieprawidłowa postać liczby.
5.	Instrukcja nie może rozpoczynać się od znaku syntaktycznego lub od liczby.
6.	Podwójnie zdefiniowana etykieta instrukcji.
7.	Opuszczony lub nieprawidłowo użyty przecinek.
8.	We wprowadzonej linii napotkano na jeden lub więcej nierozpoznanych znaków, każdy z nich zastąpiono znakiem zapytania.
9.	Nawias zamykający (prawy) nie ma odpowiadającego mu nawiasu otwierającego (lewego).
10.	Nawias otwierający (lewy) nie ma odpowiadającego mu nawiasu zamykającego (prawego) .
11.	Instrukcja usunięta ze względu na nieprawidłową interpunkcję.
12.	Przepełnienie tablicy PTPP - pojedyncza instrukcja lub definicja zagnieżdżona są za duże w stosunku do rozmiaru pamięci zarezerwowanego dla PTPP(pkt 33.5) .
13.	Instrukcja zawiera więcej niż 600 elementów. Instrukcja ta została usunięta (pkt 2.2 - definicja elementu).
14.	W programie obróbki części wystąpiło więcej definicji zmiennych niż przydzielonej dla nich pamięci.
15.	Wystąpiło przepełnienie w tablicy nazw. Pozostała część programu obróbki, część została usunięta (pkt 33.5).
16.	Instrukcja FINI pojawiła się wewnątrz makroinstrukcji lub pętli. Interpretację programu obróbki części zakończono.
17.	Słowo kluczowe języka APT zdefiniowano jako etykietę instrukcji.
18.	Zbyt duży rozmiar makroinstrukcji.
19.	W instrukcji SYN pojawiła się słowo różne od niezdefiniowanej nazwy lub słowa kluczowego.
20.	Kropka dziesiętna została użyta w nieprawidłowym kontekście.
21.	Słowo o ustalonej pozycji nie pojawiło się w kolumnach 1-6.
22.	Niedozwolona postać instrukcji DEBUG.

23. Niewłaściwa postać instrukcji FINI lub brak instrukcji FINI w programie obróbki części (została napotkana karta/*).
26. W instrukcji ZSURF po prawej stronie znaku slash nie wystąpiła nazwa płaszczyzny lub definicja płaszczyzny.
30. Instrukcja READ/2 nie może czytać innej instrukcji READ/2.
31. Nazwa podana w instrukcji READ/2 lub w instrukcji CALLwołającej makroinstrukcję systemową nie została znaleziona wśród zapamiętanych segmentów do odczytu lub makroinstrukcji systemowych.
32. Pojawił się błąd wejścia/wyjścia podczas czytania makroinstrukcji systemowej lub segmentu bądź też zbiór danych dla tego rodzaju pamięci nie może być otwarty.
34. Błąd w formacie instrukcji READ/2 - wystąpiło za dużo elementów w instrukcji.
36. Makroinstrukcja systemowa nie zawiera instrukcji TERMAC.
37. Instrukcja READ/2 nie może występować wewnątrz pętli lub makroinstrukcji.
38. Instrukcja READ/2 nie może czytać pierwszego wywołania makroinstrukcji systemowej.
40. Tablice VST i PTPF zajęły zbyt dużo pamięci (dla tablicy CANON pozostało mniej niż 1300 słów).
51. Niedozwolone wykorzystanie powierzchni.
52. Niedozwolone zastosowanie etykiety instrukcji.
53. Wyrażenie arytmetyczne zakończone jest operatorem, po którym nie następuje argument.
54. Niedozwolony znak syntaktyczny lub jego brak w miejscu, w którym wymagany jest operator arytmetyczny.
56. Zbyt wiele kolejnych potęgowań.
59. Brak nawiasów obejmujących argument obliczanej funkcji.
61. Argument w wyrażeniu arytmetycznym nie jest liczbą, zmienną lub funkcją arytmetyczną.
62. Definicja gnieźdźdźona powierzchni spowodowała przepełnienie pamięci. Należy nadać nazwę przy definiowaniu tej powierzchni.
63. Próba dzielenia przez zero.
65. Podano tylko jeden wektor jako argument funkcji wektorowej DOTF.
66. Podano niedozwolony argument dla funkcji arytmetycznej.
67. Jako argument funkcji SQRTF została podana liczba ujemna; do obliczeń wzięto wartość bezwzględna tej liczby.
68. Nie można podnosić liczby ujemnej do potęgi niecałkowitej.
70. Zbyt wiele argumentów listy zagnieźdżeń w jednej instrukcji (tzn. zagnieźdzenie zawiera więcej niż jeden parametr, lecz nie słów kluczowych głównych lub operatorów). Maksymalna dopuszczalna liczba wynosi 10.
73. Błąd w umieszczaniu zbioru danych opisujących dużą powierzchnię w rekordzie do zapisu.
74. Błąd wejścia/wyjścia podczas zapisu zbioru danych opisujących dużą powierzchnię.
76. Niedozwolone wykorzystanie słowa kluczowego.
78. Błąd formatu - niedozwolony znak syntaktyczny.
79. Błąd formatu - parametr powinien być powierzchnią.
80. Kody rodzajów powierzchni nie są takie same.

81. Pozycja jest przeznaczona dla słowa kluczowego, które się nie pojawiło.
82. Błąd formatu - na pierwszej pozycji jest nazwa, po której nie następuje znak równości.
83. Parametr nie ma wartości skalarnej.
84. Liczba przetwarzanych parametrów nie jest zgodna z długością postaci kanonicznej.
101. Niezdefiniowana nazwa.
103. Lista indeksów wystąpiła w niewłaściwym kontekście.
104. Wystąpił indeks spoza zakresu podanego w instrukcji RESERV.
105. Po nazwie, która pojawiła się w instrukcji RESERV, nie wystąpił indeks.
106. Tablica skalarna zawiera powierzchnię lub na odwrót.
108. Wystąpił indeks przy nazwie, która nie pojawiła się w instrukcji RESERV.
109. Niepoprawna postać listy indeksów.
111. Nazwa zdefiniowana jako etykieta instrukcji została podana w miejscu, gdzie powinna wystąpić powierzchnia.
129. Niepoprawna liczba użytych danych.
131. Niepoprawny format.
132. Wystąpił niewłaściwy rodzaj parametru.
133. Niedozwolony kod słowa kluczowego.
135. Zastosowano modyfikator względem powierzchni prowadzonej w instrukcji ruchu.
136. Zdefiniowana implicite powierzchnia ograniczająca została użyta w instrukcji ruchu z modyfikatorami kierunkowymi GOUP lub GODOWN.
137. Po instrukcji ruchu wykorzystującej zdefiniowaną implicite powierzchnię ograniczającą wystąpiła instrukcja GO, GODLTA lub FROM.
139. Zastosowano niedozwoloną zmienną.
152. Błąd formatu - nie został podany typ powierzchni.
153. Nie dobrano kodu typu powierzchni.
154. Wystąpiła próba uzyskania parametru leżącego poza zakresem postaci kanonicznej.
155. Błąd formatu - nie wystąpił znak slash.
156. Błąd formatu - nie wystąpiła nazwa zmiennej.
157. Wystąpiła próba zastosowania instrukcji OBTAIN do dużej powierzchni.
176. Nastąpiło przepełnienie zbioru CANON. Nie można zapisać definicji.
177. Wystąpił błąd wejścia/wyjścia przy umieszczaniu definicji w przepełnionym zbiorze CANON.
178. Zbiór został zapełniony aktywnymi makroinstrukcjami - nie ma miejsca, aby umieścić lub zapamiętać makroinstrukcję.
179. Zbiór nie może być umieszczony tak, aby odczytać makroinstrukcję lub powierzchnię.
180. Wystąpił błąd wejścia/wyjścia podczas odczytu definicji ze zbioru CANON.
181. Wystąpił błąd wejścia/wyjścia podczas odczytu z pamięci zbioru instrukcji z obszaru pętli, która miała być wykonana. Przetwarzanie pętli zostało zakończone.
201. Wystąpiła próba zdefiniowania makroinstrukcji w obszarze pętli lub innej makroinstrukcji.
202. Użyta nazwa makroinstrukcji była wcześniej zdefiniowana.
203. Po nazwie makroinstrukcji musi wystąpić znak równości, a następnie słowo MACRO i znak slash.
204. Nazwa makroinstrukcji nie jest poprawną nazwą.

205. W instrukcji MACRO lub CALL wystąpiło więcej niż 50 zmiennych.
206. Wewnątrz pętli wystąpiła instrukcja LOOPST.
207. W instrukcji MACRO po znaku slash wystąpiła niewłaściwa wielkość.
208. Wystąpiła instrukcja LOOPND bez odpowiadającej jej instrukcji LOOPST.
209. Wystąpiła instrukcja TERMAC poza definicją makroinstrukcji.
210. Wewnątrz makroinstrukcji wystąpiła instrukcja LOOPST lub LOOPND.
211. Niewłaściwie użyta nazwa makroinstrukcji.
212. Nie znaleziono instrukcji LOOPND podczas wykonywania instrukcji z obszaru pętli; jest to prawdopodobnie spowodowane błędnym zdefiniowaniem etykiety instrukcji LOOPND.
214. Wystąpił niewłaściwy znak syntaktyczny przy określaniu wartości normalnej zmiennej makro w instrukcji MACRO. Dozwolone jest tylko podanie znaku + lub - przed liczbą (a nie nazwą zmiennej).
226. Błędny rodzaj słowa po znaku slash w instrukcji CALL.
227. Po słowie CALL nie wystąpił znak slash.
228. Nazwa zmiennej makro podana w instrukcji CALL nie wystąpiła w instrukcji MACRO.
229. Po nazwie zmiennej w instrukcji CALL nie wystąpił znak równości.
230. Nie nadano wartości jednej ze zmiennych makro.
231. Poziom zagnieżdżenia się makroinstrukcji jest większy od 5.
232. Niedozwolone zagnieżdżanie lub zmienna indeksowana w instrukcji CALL.
233. Nieprawidłowo zdefiniowana nazwa makroinstrukcji następująca po CALL / .
234. W instrukcji CALL pojawił się niewłaściwy znak syntaktyczny - powinien wystąpić przecinek.
235. Zastosowano niewłaściwy znak syntaktyczny przy nadawaniu wartości zmiennej makro w instrukcji CALL; może być zaakceptowany tylko znak plus lub minus poprzedzający liczbę lecz nie nazwę zmiennej.
251. Nazwa, która pojawiła się w instrukcji jako nazwa zmiennej lub elementu geometrycznego jest etykietą instrukcji.
252. Instrukcja RESERV nie może pojawić się wewnątrz pętli lub makroinstrukcji.
253. Błędny format - w instrukcji pojawiła się niedozwolona pozycja. Może być to słowo kluczowe, nazwa definiowanej powierzchni, etykieta instrukcji, znak syntaktyczny lub pozycja niezidentyfikowana.
254. Błędny format - oczekiwano pojawienia się nazwy zmiennej prostej lub nazwy zarezerwowanego pola.
255. Błędny format - pojawiła się niezdefiniowana nazwa, podczas gdy oczekiwano, że pojawi się wartość.
256. Nazwa była już wcześniej używana przed pojawieniem się w instrukcji RESERV.
272. Brak miejsca w pamięci dla listy argumentów lub listy indeksów.
274. Niepoprawny format lub zły typ danych wejściowych dla listy indeksów.
275. Podano niepoprawną liczbę nazw w zagnieżdżonej liście indeksów.
302. Pojawił się błąd podczas próby wyprowadzenia rekordu pośredniego (PTPP).
303. Pojawił się błąd wejścia/wyjścia podczas wydruku zbioru używanego do znajdowania powierzchni ograniczającej zdefiniowany implicite.

- 304. Pojawił się błąd wejścia/wyjścia podczas odczytu lub zapisu zbioru używanego do znajdowania powierzchni ograniczającej zdefiniowanej *implicit*.
- 351. Zła liczba danych w instrukcji SYN.
- 353. Instrukcja SYN nie może pojawić się w makroinstrukcji.
- 354. Przepełnienie tablicy synonimów.
- 355. Program oczekuje na pojawienie się nazwy w instrukcji SYN.
- 356. Program oczekuje na pojawienie się słowa kluczowego w instrukcji SYN.
- 376. Błąd w formacie instrukcji POCKET - osiem pierwszych argumentów następujących po znaku slash nie jest skalarami.
- 377. Niepoprawna liczba danych opisujących punkty ograniczające wybrania ulżeniowego.
- 378. Przekroczenie roboczego obszaru pamięci PROTP. Przetwarzana instrukcja próbuje utworzyć większy rekord zbioru PROTAP, niż sam system może przetworzyć.
- 402. W instrukcji IF w nawiasach nie podano liczby, wyrażenia lub nazwy uprzednio zdefiniowanej zmiennej skalarnej.
- 403. W instrukcjach IF lub JUMPTO nie wystąpiła etykieta.
- 404. Nastąpiło przepełnienie tablicy zmiennych podczas przetwarzania etykiety instrukcji (w celu poprawienia błędu - zob. instrukcja TUNEUP).
- 406. Etykieta nie występuje w tym samym obszarze pętli lub makroinstrukcji co instrukcja IF lub JUMPTO.
- 407. Instrukcja IF lub JUMPTO może być używana tylko wewnątrz pętli lub makroinstrukcji.
- 408. Błąd systemu - indeks tablicy PTPP przekracza dopuszczalny zakres.
- 409. Niewłaściwa liczba etykiet w instrukcji IF, JUMPTO lub TRANTO.
- 453. Na pozycji etykiety instrukcji nie wystąpiła nazwa etykiety, liczba lub niezdefiniowana nazwa.
- 456. Wystąpiła próba przeniesienia do instrukcji, która nie generuje rekordu PROTAP.
- 501. Niewłaściwe wykorzystanie powierzchni w instrukcji postprocesora.
- 503. Użyto niewłaściwego rodzaju parametru w instrukcji postprocesora na prawo od znaku slash.
- 504. Wołanych było zbyt wiele postprocesorów.
- 505. Podano parametry w instrukcji postprocesora, która nie wymaga parametrów.
- 506. Niepoprawna postać instrukcji TRACLT.
- 507. Nieprawidłowa liczba parametrów w instrukcji COPY.
- 553. W instrukcji PTONLY po znaku slash występuje niewłaściwa pozycja.
- 576. Nieprawidłowa postać instrukcji PSIS lub ZSURF.
- 650. Zamiast oczekiwanej powierzchni ograniczającej zdefiniowanej *implicit* pojawiła się instrukcja FINI.
- 651. W instrukcji REFSYS wystąpił inny parametr niż słowo kluczowe NOMORE lub definicja macierzy.
- 701. Pierwsze dwa punkty w definicji powierzchni prostokątnej są położone zbyt blisko siebie (w odległości mniejszej niż 0.032).
- 702. Nie można zdefiniować płaszczyzny na podstawie definicji powierzchni prostokątnej - trzy punkty są prawie współliniowe.

800. Błąd w formacie definicji rozkładu punktów.
801. W równoległym rozkładzie punktów można zastosować tylko liniowy rozkład punktów.
802. Przepełnienie obszaru pamięci; rozkład punktów jest za duży (przekracza dopuszczalną liczbę 333 punktów).
811. Operacje OMIT i RETAIN zostały podane w tej samej instrukcji programu obróbki części. Tylko jeden typ operacji może wystąpić w podanej instrukcji, chociaż może być używana dowolna liczba operacji tego samego typu.
812. Liczba identyfikująca punkt dla operacji OMIT, RETAIN i AVOID wynosi zero, jest ujemna lub większa od liczby punktów w rozkładzie punktów.
820. Definiowany rozkład punktów musi mieć nazwę.
830. W instrukcji GOTO po nazwie rozkładu punktów pojawiła się liczba, zanim wystąpił modyfikator OMIT, RETAIN lub AVOID.
831. Po nazwie rozkładu punktów w instrukcji GOTO pojawił się inny element niż nazwa zmiennej lub dopuszczalne słowo kluczowe.
832. Po nazwie rozkładu punktów w instrukcji GOTO wystąpiło niedozwolone słowo kluczowe.
833. W instrukcji GOTO, po nazwie rozkładu punktów, wystąpił modyfikator ZIGZAG lub INVERS po modyfikatorze OMIT, RETAIN lub AVOID.
834. Wystąpił więcej niż jeden modyfikator INVERS lub ZIGZAG w tej samej instrukcji.
835. Modyfikator ZIGZAG został podany wraz z rozkładem punktów, który nie był zdefiniowany jako równoległy (PARLEL).
836. Niewystarczające lub niepoprawne dane wystąpiły po modyfikatorach OMIT, RETAIN, AVOID lub THRU.
837. Nie wystąpiły ograniczenia na ciąg THRU w tej samej kolumnie rozkładu równoległego dla modyfikatora ZIGZAG.
838. Liczba punktów wołanych przez operacje OMIT, RETAIN lub AVOID w danej chwili przekroczyła liczbę określoną na liście modyfikatorów lub dopuszczalną dla tej operacji - wskazuje to na częściowe pokrywanie się operacji lub ich niewłaściwą kolejność.
- Błąd ten pojawia się też wtedy, gdy operacja AVOID została określone względem punktu, który nie pojawia się w wyjściowym ciągu punktów z rozkładu punktów (punkt był usunięty przez operacje RETAIN lub OMIT).
840. Pojawił się błąd wejścia/wyjścia podczas poszukiwania żądanej definicji rozkładu punktów w zbiorze pamięci.
841. Pojawił się błąd wejścia/wyjścia podczas czytania żądanej definicji rozkładu punktów ze zbioru pamięci.
1000. Niedozwolony format dla tego typu powierzchni.
1001. Niepoprawna długość danych dla tego sposobu definiowania.
1002. Błąd formatu, nieznan typ powierzchni lub dwa razy zdefiniowana zmienna.
1003. W definicji użyto złego modyfikatora.
1005. Powierzchnia geometryczna jest matematycznie niezdefiniowana dla potrzeb dostarczonego opisu.
1006. Typ powierzchni jest niezgodny z żądanym w danym kontekście.
1550. Podano za mało parametrów w definicji.

- 1600. Wewnątrz makroinstrukcji wystąpiła instrukcja FINI.
- 1601. W obszarze pętli wystąpiła instrukcja FINI.
- 2000. Błąd formatu wejścia/wyjścia.
- 2001. Nazwa wprowadzona przez instrukcję READ przeddefiniowuje powierzchnię na zmienną lub na odwrót.
- 2002. Zbyt mało kart danych dla skompletowania listy czytanych argumentów.
- 2004. Podany na karcie indeks przekracza obszar aktualnie zarezerwowanego w instrukcji RESERV pola.
- 2005. Wystąpił parametr bez nazwy na liście PUNCH .
- 6102. Błąd systemu APT.
- 6110. Błąd systemu APT.
- 6113. Błąd systemu APT.
- 6128. Błąd systemu APT.
- 6133. Błąd systemu APT.
- 6134. Błąd systemu APT.
- 6153. Błąd systemu APT.
- 6214. Błąd systemu APT.
- 12345. Przywołaniu programu specjalnego instrukcją CALL bezpośrednio po znaku slash wystąpił element różny od nazwy zmiennej.

D O D A T E K C

Komunikaty o błędach wykrywanych w fazie pracy jednostki arytmetycznej

W poniższym spisie błędy o numerach poprzedzonych znakiem minus są błędami ostrzegawczymi. Oznacza to, że po pojawieniu się tego błędu i wydrukowaniu odpowiedniego komunikatu, system APT kontynuuje przetwarzanie programu obróbki części.

W zamieszczonej poniżej liście podawana jest w nawiasach nazwa programu, który wykrywa ten błąd - informacja ta może pomóc w zlokalizowaniu przyczyny błędu.

2. Niedozwolony rekord danych PROTAP dla instrukcji TLAXIS - czwarte słowo rekordu PROTAP powinno zawierać 1,2,3,4 lub 5. Należy zwrócić się do programisty systemowego.
3. Nie można znormalizować wektora podanego w instrukcji TLAXIS (ASEC2). Należy sprawdzić wektor podany w instrukcji TLAXIS - przynajmniej jedna składowa wektora musi mieć wartość różną od zera. Jeżeli błąd powtarza się, należy zwrócić się do programisty systemowego.
4. Niedozwolona pozycja czwartego parametru w rekordzie PROTAP dla instrukcji NORMPS lub NORMDS (ASEC2) . Należy zwrócić się do programisty systemowego.
5. Niedozwolona dla instrukcji TLAXIS/ATANGL pozycja czwartego parametru w rekordzie PROTAP (ASEC2). Należy zwrócić się do programisty systemowego.
6. Zerowa wartość w instrukcji INDIRP (ASEC2). Aktualne położenie narzędzia pokrywa się z punktem podanym w instrukcji INDIRP.
7. Zerowy wektor w instrukcji INDIRV(ASEC2). Przynajmniej jedna składowa wektora zdefiniowanego

w instrukcji INDIRV musi mieć wartość niezerową.

8. Zerowy wektor w instrukcji SRFVCT(ASEC2). Przynajmniej jedna składowa wektora zdefiniowanego przez instrukcję SRFVCT musi mieć wartość niezerową.
- 9. Ostrzeżenie - podczas przetwarzania instrukcji TRANTO lub wielokrotnej powierzchni ograniczającej pojawiła się instrukcja FINI(ASEC2 ... A 2030000).
Etykiety, do której odwołują się instrukcje TRANTO lub wielokrotna powierzchnia ograniczająca muszą pojawiać się przy instrukcjach generujących informacje dla jednostki arytmetycznej, fazy edycji lub fazy pracy postprocesora. Zawierają one co następuje.
 - Instrukcje przesuwu wstępnego i instrukcje ruchu (GO, OFFSET, GOLFT, GORGT, itd.).
 - Informacje odnoszące się do ruchu narzędzia (INTOL, OUTTOL, AUTOPS, CUTTER, TLAXIS, itp.).
 - Informacje używane w fazie edycji (TRACUT, COPY, itd.).
 - Informacje wykorzystywane przez postprocesor (COOLNT, SPINDL, TOOLNO, itd.).
10. Wystąpił błąd wejścia/wyjścia podczas odczytu zbioru danych PROTAP(ASEC2). Należy zwrócić się do programisty systemowego.
11. Wystąpił błąd wejścia/wyjścia podczas zapisu zbioru danych CLTAPE(ASEC2). Należy zwrócić się do programisty systemowego.
12. Pojawił się błąd wejścia/wyjścia w czasie wykonywania się podprogramu ASERCH(ASEC2). Należy zwrócić się do programisty systemowego.
13. Wystąpił błąd wejścia/wyjścia podczas odczytu zbioru danych opisujących dużą powierzchnię (ASEC2). Należy zwrócić się do programisty systemowego.
16. Kąt α w instrukcji TLAXIS/ATANGL, n, α jest co do modułu większy od 90° (ASEC2). Kąt α musi być podany w stopniach, zaś jego wartość musi znajdować się w zakresie od -90° do $+90^\circ$.
17. Kąt β w instrukcji
 TLAXIS/ATANGL, n, α , CUTANG, β
 jest co do modułu większy od 90° (ASEC2). Kąt β musi być podany w stopniach, zaś jego wartość musi znajdować się w zakresie od -90° do $+90^\circ$.
18. Wystąpiła niedozwolona dla instrukcji TLAXIS/ATANGL liczba słów w zbiorze danych PROTAP (ASEC2). Należy zwrócić się do programisty systemowego.
19. Podany w instrukcji
 TLAXIS/ATANGL, n, α , WE
 wektor WE nie nadaje się do normalizacji (ASEC2). Przynajmniej jedna składowa wektora WE powinna mieć wartość niezerową.
30. Wystąpił błąd przy przetwarzaniu instrukcji OFFSET(AMOVE). Należy sprawdzić, czy narzędzie jest odpowiednio ustawione względem powierzchni prowadzącej. Pierwszą powierzchnią na prawo od znaku slash jest powierzchnia prowadząca.
31. Wystąpił błąd przy przetwarzaniu instrukcji OFFSET(AMOVE). Należy sprawdzić, czy narzędzie jest prawidłowo ustawione względem powierzchni prowadzącej. Jeżeli błąd powtarza się, należy się zwrócić do programisty systemowego.
33. Wystąpił błąd przy przetwarzaniu instrukcji OFFSET (AOFSET). Instrukcja OFFSET powinna być poprzedzona instrukcją INDIRV lub INDIRP, poprawnie definiującą wektor.

34. Wystąpił błąd przy przetwarzaniu instrukcji OFFSET (AOFSET). Wektor określony przez instrukcję INDIRV lub INDIRP nie przebija powierzchni prowadzącej lub wektor osi narzędzia nie przebija powierzchni przedmiotu.
35. Wystąpił błąd przy przetwarzaniu instrukcji OFFSET (AOFSET). Błąd systemu - należy się porozumieć z programistą systemowym.
36. Podano niewłaściwą powierzchnię przedmiotu w instrukcji OFFSET.
- 37. Ostrzeżenie - występują trudności z ustawieniem narzędzia względem powierzchni przedmiotu zgodnie z wymaganiami podanymi w instrukcji OFFSET (AOFSET).
38. Wystąpił błąd przy przetwarzaniu instrukcji OFFSET (AOFSET). Należy sprawdzić wektor określony przez instrukcje INDIRV lub INDIRP.
39. Wystąpił błąd przy przetwarzaniu instrukcji OFFSET (AOFSET). Instrukcje INDIRV lub INDIRP definiują wektor styczny do powierzchni względnie wektor nie przebija powierzchni prowadzącej. Pierwszą powierzchnią podaną po znaku slash jest powierzchnia prowadząca.
40. Błąd podczas przesuwu wstępnego (instrukcja GO), gdy system pracuje w trybie 2D (ASTUP2). Narzędzie jest ustawione prawidłowo na powierzchni - należy zastosować instrukcję INDIRV lub INDIRP.
41. Niedozwolony skok w module ASEC2 ... - należy się porozumieć z programistą systemowym (ASEC2).
42. Błąd podczas przesuwu wstępnego (instrukcja GO) z instrukcją NOPS, gdy system pracuje w trybie 2D (ASTUP). Poprawę może przynieść zastosowanie trybu pracy 3DCALC.
43. Błąd podczas przesuwu wstępnego (instrukcja GO), gdy system pracuje w trybie 2D (ASTUP2). Poprawę może przynieść zastosowanie trybu pracy 3DCALC.
44. Błąd podczas przesuwu wstępnego (instrukcja GO), gdy system pracuje w trybie 2D (ASTUP2). Poprawę może przynieść zastosowanie trybu pracy 3DCALC.
302. Przekroczono dopuszczalną liczbę powierzchni ograniczających (ASEC2).
303. Przekroczono obszar pamięci przeznaczony dla powierzchni (ASEC2). Należy się porozumieć z programistą systemowym.
- 307. Ostrzeżenie - ciąg wektorów drogi narzędzia pojawił się w zbiorze danych PROTAP (ASEC2). Należy się porozumieć z programistą systemowym. Prawdopodobnie została zamontowana niewłaściwa taśma dla zbioru PTONLY lub utworzony nieprawidłowy zbiór PROTAP przez moduł 1.
311. Narzędzie nie było zdefiniowane przed przesuwem wstępnym (ASEC2).
312. Narzędzie nie było zdefiniowane przed podaniem instrukcji ruchu (ASEC2).
314. Niedozwolony typ kodu 13 000 w zbiorze danych PROTAP (ASEC2). Prawdopodobnie zamontowano niewłaściwą taśmę dla zbioru PTONLY lub utworzono zły zbiór PROTAP przez moduł 1. Należy się porozumieć z programistą systemowym.
315. Nie znaleziono poprawnej etykiety podczas przetwarzania instrukcji TRANTO lub instrukcji ruchu z wielokrotną powierzchnią ograniczającą (ASEC2 ... A2030000).
 - Należy sprawdzić możliwość niewłaściwej struktury logicznej pętli (instrukcje IF, JUMPTO) rozpatrywanej przez moduł 1 systemu lub struktury logicznej pętli (instrukcja TRANTO, wielokrotna powierzchnia ograniczająca) rozpatrywanej przez moduł 2 systemu.
 - Należy sprawdzić rozmieszczenie etykiet instrukcji. Etykiety, do których odwołuje się

instrukcja TRANTO lub wielokrotna powierzchnia ograniczająca, muszą być umieszczone przy instrukcjach, które generują informacje dla jednostki arytmetycznej (moduł 2 systemu APT), edytora (moduł 3 systemu APT) lub postprocesora (zob. komunikat ostrzegawczy nr - 9).

401. Średnica narzędzia jest liczbą ujemną (ATLSEG).
402. Wysokość narzędzia jest liczbą ujemną (ATLSEG).
403. Kąt narzędzia ($\alpha + \beta$) jest za duży (ATLSEG).
404. Kąt narzędzia α jest ujemny (ATLSEG).
405. Średnica narzędzia, kąt α i wysokość narzędzia są niekonsekwentne (ATLSEG).
406. Kąt narzędzia β jest ujemny (ATLSEG).
407. Promień łuku narożnego narzędzia, gdzie termin "łuk narożny" odnosi się do zaokrąglenia, jest ujemny (ATLSEG).
408. Środek łuku narożnego jest podany niekonsekwentnie w stosunku do górnego odcinka narzędzia (ATLSEG).
409. Narożny łuk narzędzia przecina dolny odcinek narzędzia pod niedozwolonym kątem (ATLSEG).
410. Narożny łuk narzędzia nie przecina dolnego odcinka narzędzia (ATLSEG).
411. Narożny łuk narzędzia nie przecina górnego odcinka narzędzia (ATLSEG).
412. Narożny łuk narzędzia przecina górny odcinek narzędzia pod niedozwolonym kątem (ATLSEG).
413. Brak górnego odcinka narzędzia - łuk narożny narzędzia styka się z dolnym odcinkiem w nieprawidłowy sposób (ATLSEG).
414. Nie zdefiniowano odcinków narzędzia (ATLSEG).
- 415. Ostrzeżenie - brak dolnej linii narzędzia; kąt okręgu stycznego do osi narzędzia nie jest równy α (ATLSEG).
- 416. Ostrzeżenie - brak górnej linii narzędzia; kąt okręgu stycznego w górnym punkcie nie jest równy β (ATLSEG).
417. Niedozwolony kształt narzędzia - średnica narzędzia jest podana niekonsekwentnie w stosunku do promienia zaokrąglenia.
502. Narzędzie nie znajduje się w paśmie tolerancji powierzchni przedmiotu przed rozpoczęciem ruchu roboczego (AREPRE).
 - Narzędzie musi być prawidłowo ustawione, zarówno względem powierzchni prowadzącej, jak i powierzchni przedmiotu przed rozpoczęciem ruchu roboczego (opisanego za pomocą instrukcji ruchu). Należy sprawdzić, czy procedura przesuwu wstępnego lub instrukcje ruchu poprzedzające aktualny ruch narzędzia, ustawiają narzędzie w poprawny sposób względem powierzchni przedmiotu dla potrzeb bieżącej instrukcji ruchu. Należy też sprawdzić aktualne ustawienie narzędzia względem powierzchni przedmiotu i stwierdzić, czy jest ono zgodne z podanymi w instrukcji ruchu modyfikatorami pozycyjnymi.
503. Narzędzie nie znajduje się w paśmie tolerancji powierzchni prowadzącej przed rozpoczęciem ruchu roboczego (AREPRE).
 - Narzędzie musi być prawidłowo ustawione, zarówno względem powierzchni prowadzącej, jak i powierzchni przedmiotu przed rozpoczęciem ruchu roboczego. Należy sprawdzić, czy procedura przesuwu wstępnego lub instrukcje ruchu poprzedzające aktualny ruch narzędzia, ustawiają narzędzie w poprawny sposób względem powierzchni

przewodzącej.

Należy też sprawdzić aktualne ustawienie narzędzia względem powierzchni prowadzącej i stwierdzić, czy jest ono zgodne z podanymi w instrukcji ruchu modyfikatorami pozycyjnymi dotyczącymi usytuowania narzędzia względem powierzchni prowadzącej.

504. Nie można określić kierunku ruchu "do przodu" (AFWDSL). Jeszcze przed wystąpieniem instrukcji ruchu (GOLFT, GORGT, itp.) narzędzie musi mieć określone położenie w układzie współrzędnych oraz musi być zdefiniowane pojęcie kierunku.
505. Nie można określić kierunku ruchu (ATISEL). Należy sprawdzić podany modyfikator kierunkowy (GOLFT, GORGT itp.).
601. W obrębie działania instrukcji NOPS nie użyto instrukcji przesuwu wstępnego względem jednej powierzchni (ASTRUP).
602. Błąd przy przetwarzaniu instrukcji przesuwu wstępnego w postaci GO(ASTRUP).
- Przed instrukcją przesuwu wstępnego można podać instrukcje INDIRV lub INDIRP, w których podane wektory wskażą prawidłowy kierunek do powierzchni.
 - Należy sprawdzić, czy możliwe jest podane w instrukcji ustawienie narzędzia względem powierzchni.
603. Niedozwolone dane PROTAP dla instrukcji przesuwu wstępnego (ASTRUP). Należy porozumieć się z programistą systemowym.
701. Nie można określić kierunku "do przodu" (ARLM2). System APT wybrał metodę analityczną (2D) obliczania drogi narzędzia.
- Przed podaniem instrukcji ruchu (GOLFT, GORGT itp.) narzędzie musi mieć określone położenie w układzie współrzędnych oraz musi być określony kierunek ruchu.
703. Powierzchnia prowadząca nie jest prostą ani okręgiem (ADSCF).
- System APT wybrał metodę analityczną (2D) obliczania drogi narzędzia. Można spróbować zastosować tryb pracy 3DCALC.
704. Płaszczyzna prowadząca jest równoległa do płaszczyzny OXY (ADSCF).
- System APT wybrał metodę analityczną (2D) obliczania drogi narzędzia. Można spróbować zastosować tryb pracy 3DCALC.
706. Powierzchnia ograniczająca nie jest prostą lub okręgiem (ACSCF).
- System APT wybrał metodę analityczną (2D) obliczenia drogi narzędzia. Można spróbować zastosować tryb pracy 3DCALC.
707. Płaszczyzna ograniczająca jest równoległa do płaszczyzny OXY (ACSCF).
- System APT wybrał analityczną metodę (2D) obliczania drogi narzędzia. Można spróbować zastosować tryb pracy 3DCALC.
- 708. Ostrzeżenie - promień okręgu, będącego powierzchnią ograniczającą, jest mniejszy od promienia narzędzia (ACSCF).
- System APT wybrał analityczną metodę (2D) obliczania drogi narzędzia. Może być wybrany niewłaściwy punkt zatrzymania się narzędzia. Można spróbować zastosować tryb pracy 3DCALC.
712. Poprawka, wynikająca z zastosowania instrukcji THICK, powoduje, że promień okręgu będącego powierzchnią ograniczającą, staje się ujemny (ATECOR).
713. Powierzchnia prowadząca lub powierzchnia ograniczająca nie są prostymi lub okręgami (AINT).

- System APT wybrał analityczną metodę 2D obliczania drogi narzędzia. Można spróbować zastosować tryb pracy 3DCALC.
714. Nie znaleziono akceptowanej pozycji zatrzymania narzędzia (AINT).
System APT wybrał analityczną metodę (2D) obliczania drogi narzędzia. Należy sprawdzić wzajemne położenie powierzchni prowadzącej i ograniczającej oraz położenie narzędzia względem powierzchni prowadzącej.
715. Powierzchnia prowadząca i ograniczająca są liniami równoległymi, więc nie istnieje akceptowalna pozycja zatrzymania narzędzia (AINT).
System APT wybrał analityczną metodę (2D) obliczania drogi narzędzia.
717. Niewłaściwe wzajemne położenie powierzchni prowadzącej i ograniczającej (ACCINT).
718. Nie można określić końcowej pozycji zatrzymania narzędzia w instrukcji ruchu, gdzie powierzchnia prowadząca jest okręgiem, powierzchnia przedmiotu jest płaszczyzną, zaś powierzchnia ograniczająca jest okręgiem (ACCINT).
System APT wybrał analityczną metodę (2D) obliczania drogi narzędzia. Można spróbować zastosować tryb pracy 3DCALC.
720. Modyfikator TANTO użyte w sytuacji prosta - prosta (AINT).
721. Podany został modyfikator TANTO, chociaż nie istnieje punkt styczności (AINT).
724. Żaden ze znalezionych punktów nie spełnia wymagań dla punktu końcowego (ASELP).
- 725. Ostrzeżenie - wystąpiły trudności podczas określania pozycji zatrzymania narzędzia w instrukcji ruchu, gdzie powierzchnia prowadząca jest okręgiem, powierzchnia przedmiotu jest płaszczyzną, zaś powierzchnia ograniczająca jest okręgiem (ACCINT).
System APT wybrał analityczną metodę (2D) obliczania drogi narzędzia. Można spróbować zastosować tryb pracy 3DCALC.
725. Płaszczyzna przedmiotu jest prawie prostopadła do płaszczyzny OXY (AZCOR). System APT wybrał analityczną metodę (2D) obliczeń drogi narzędzia. Można spróbować zastosować tryb pracy 3DCALC.
730. Promień narzędzia jest równy zeru, zaś wymagane jest ustawienie narzędzia spełniające warunki TO lub PAST (AINT).
740. Błąd systemu podczas wyboru punktu w trybie pracy 2D. Należy się porozumieć z programistą systemowym.
801. Nie można określić długości kroku wzdłuż powierzchni prowadzącej i powierzchni przedmiotu podczas realizacji instrukcji ruchu, gdy jednostka arytmetyczna pracuje w trybie 3D (ARLM3). Należy sprawdzić ustawienie narzędzia względem powierzchni.
802. Błąd przy określaniu końcowej pozycji zatrzymania narzędzia podczas realizacji instrukcji ruchu, gdy jednostka arytmetyczna pracuje w trybie 3D.
Należy sprawdzić, czy są spełnione warunki na poprawne ustawienie narzędzia względem powierzchni ograniczającej.
803. Wyczerpany licznik NUMPTS, podczas realizacji instrukcji ruchu, gdy jednostka arytmetyczna pracuje w trybie 3D (ARLM3).
• Należy sprawdzić poprawność modyfikatorów kierunkowych (GOLFT, GORGT itp.) - podany kierunek może wskazywać kierunek przeciwny w stosunku do powierzchni ograniczającej.
• Jeżeli kierunek jest poprawny, należy zwiększyć licznik NUMPTS.
804. Wystąpił błąd podczas próby przesunięcia wzdłuż powierzchni prowadzącej i powierzchni przed-

miotu, gdy jednostka arytmetyczna pracuje w trybie 3D (ARLM3).

Należy zmniejszyć długość kroku (instrukcja MAXDP) przed wykonaniem instrukcji ruchu - w przybliżeniu około $3/4$ normalnej długości kroku dla powierzchni przedmiotu lub powierzchni prowadzącej (w zależności od tego, która ma większą krzywiznę). Należy również odpowiednio zmienić licznik punktów (instrukcja NUMPTS).

- 805. Ostrzeżenie - wystąpiła trudność w określeniu akceptowalnej długości kroku dla instrukcji ruchu, gdy jednostka arytmetyczna pracuje w trybie 3D (ARLM3).
- 821. Ostrzeżenie - wystąpiła trudność w określeniu odległości od aktualnego położenia narzędzia do jego pozycji zatrzymania, stycznej do powierzchni ograniczającej (ACHECK).
- 822. Ostrzeżenie - system wykrył, że narzędzie odsuwa się od powierzchni ograniczającej (ACHECK).
- 823. Ostrzeżenie - wystąpiła trudność w określeniu prawidłowej orientacji osi narzędzia (ACHECK).
- 841. Podczas wykonania instrukcji ruchu licznik punktów NUMPTS został wyczerpany zarówno w odniesieniu do płaszczyzny prowadzącej jak i ograniczającej (APLANE).
 - Należy sprawdzić modyfikator kierunkowy (GOLFT, GORGT itp.). Określony w instrukcji kierunek może wskazywać kierunek przeciwny względem powierzchni ograniczającej.
 - Jeżeli podany kierunek jest prawidłowy, wówczas należy zwiększyć licznik punktów NUMPTS i ewentualnie długość kroku (instrukcja MAXDP).
- 842. Nie można określić końcowej pozycji zatrzymania narzędzia przy wykonaniu instrukcji ruchu, gdzie zarówno powierzchnia przedmiotu jak i powierzchnia ograniczająca są płaszczyznami (APLANE).
- 843. Nie można określić końcowej pozycji zatrzymania narzędzia przy wykonaniu instrukcji ruchu, gdzie zarówno powierzchnia przedmiotu, jak i powierzchnia prowadząca są płaszczyznami (APLANE).
- 845. Ostrzeżenie - znaleziony przy wykonaniu instrukcji ruchu punkt końcowy nie znajduje się w kierunku "do przodu" w sytuacji, gdzie zarówno powierzchnia przedmiotu, jak i powierzchnia prowadząca są płaszczyznami (APLANE).
- 846. Ostrzeżenie - wystąpiła trudność w określeniu końcowej pozycji zatrzymania narzędzia przy wykonaniu instrukcji ruchu, gdzie zarówno powierzchnia przedmiotu, jak i powierzchnia prowadząca są płaszczyznami (APLANE).
Należy sprawdzić, czy narzędzie jest prawidłowo ustawione względem powierzchni ograniczającej.
- 853. Niemożliwa do spełnienia specyfikacja kąta w instrukcji TLAXIS dla układu 4-osiowego.
- 859. Niemożliwa do spełnienia specyfikacja kąta w instrukcji TLAXIS dla układu 5-osiowego.
- 881. Ostrzeżenie - napotkano na trudności przy analizie instrukcji GOUGCK (AGOUGCK).
- 882. Ostrzeżenie - wystąpiła trudność w określeniu akceptowalnego wektora drogi narzędzia przy analizie instrukcji GOUGCK (AGOUGCK).
- 901. Ostrzeżenie - wystąpiła trudność w ustawieniu narzędzia względem podanej powierzchni, co spowodowało trudność w określeniu następnego punktu (ACENTR).
- 902. Nie można prawidłowo ustawić narzędzia względem powierzchni przedmiotu i powierzchni prowadzącej lub względem powierzchni przedmiotu, powierzchni prowadzącej i powierzchni ograniczającej, ponieważ wystąpiła trudność w obliczeniu pozycji narzędzia względem powierzchni.

Należy sprawdzić, czy i tnieje prawidłowe ustawienie narzędzia względem powierzchni (ACENTR).

903. Nie można prawidłowo ustawić narzędzia względem powierzchni przedmiotu i powierzchni prowadzącej pracy w układzie 5-osiowym.
904. Nie można ustalić nowej osi narzędzia podczas obłożeń w układzie 5-osiowym (ACENTR).
- 905. Ostrzeżenie - wystąpiła trudność w ustawieniu narzędzia względem powierzchni lub w obłożeniu następnego punktu (ACNTR2).
- 906. Ostrzeżenie - nie można określić pozycji zatrzymania narzędzia stycznej do powierzchni ograniczającej (ACNTR2).
907. Wyczerpano zawartość licznika przy pracy w układzie 5-osiowym przy ponownym starcie (ACENTR).

Należy sprawdzić specyfikację osi narzędzia w instrukcji

$$TLAXIS/ATANGL, \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \alpha, \begin{bmatrix} \frac{WE}{CUTANG, \beta} \end{bmatrix}$$

oraz ustawienie narzędzia względem powierzchni.

- 910. Ostrzeżenie - wystąpiła trudność w ustaleniu końcowej pozycji zatrzymania narzędzia w wypadku styczności lub pozornej styczności (ACNTR2).
1002. Wystąpiła trudność obliczenia ustawienia narzędzia względem powierzchni (AMIND).
- Należy sprawdzić, czy istnieje właściwe ustawienie narzędzia względem powierzchni.
 - W przypadku przesuwu wstępnego, należy zastosować instrukcję INDIRV lub INDIRP w celu wskazania właściwej powierzchni.
 - W przypadku instrukcji ruchu należy zmniejszyć długość kroku (instrukcja MAXDP) - około 3/4 normalnej długości kroku dla powierzchni przedmiotu lub powierzchni prowadzącej w zależności od tego, która ma większą krzywiznę. Licznik punktów NUMPTS musi być odpowiednio zmieniony.
- 1004. Ostrzeżenie - wystąpiła trudność przy ustawieniu w jednej linii prostopadłej do narzędzia i prostopadłej do powierzchni (AMIND). W związku z tym trudno ustawić narzędzie względem powierzchni. Należy się porozumieć z programistą systemowym.
- 1008. Ostrzeżenie - naruszono konwencję JTN. Wystąpiła trudność w ustawieniu narzędzia względem powierzchni. Należy się porozumieć z programistą systemowym.
1009. Próba naruszenia konwencji JTN nie powiodła się. Wystąpiła trudność w ustawieniu narzędzia względem powierzchni. Należy się porozumieć z programistą systemowym.
1021. Nie można obliczyć prostopadłej w punkcie powierzchni (AJUNDD). Wystąpiła trudność w ustawieniu narzędzia względem powierzchni. Należy się porozumieć z programistą systemowym.
- 1022. Ostrzeżenie - wybrany początkowo kierunek nie przecina powierzchni (AJUNDD). Wystąpiła trudność w ustawieniu narzędzia względem powierzchni. Należy się porozumieć z programistą systemowym.
1041. Nie można obliczyć wektora U1 rozpoczynając z dwóch lub większej liczby kierunków (AU1CMP).
- Wystąpiła trudność w ustawieniu narzędzia względem powierzchni.
 - Należy się porozumieć z programistą systemowym.

1061. Punkt odniesienia narzędzia REFPT i środek powierzchni CC pokrywają się (ATLNRM).
- Wystąpiła trudność w ustawieniu narzędzia względem powierzchni.
 - Należy się porozumieć z programistą systemowym.
- 1081. Ostrzeżenie - wystąpiła trudność w zlokalizowaniu powierzchni.
- Napotkano na trudność przy obliczeniu ustawienia narzędzia względem powierzchni.
 - Należy umieścić przed instrukcją przesuwu wstępnego instrukcję INDIRV lub INDIRP, w których podano wektory prawidłowo wskażą powierzchnię.
 - Należy sprawdzić, czy istnieje prawidłowe ustawienie narzędzia względem powierzchni.
1082. Wektor prostopadły do narzędzia TN jest zerowy (ATLNRM). Wystąpiła trudność w ustawieniu narzędzia względem powierzchni. Należy się porozumieć z programistą systemowym.
1083. Nie można znaleźć kierunku, który przecina powierzchnię (ATLNRM).
- Wystąpiła trudność w ustawieniu narzędzia względem powierzchni.
 - Należy umieścić przed instrukcją przesuwu wstępnego instrukcję INDIRV lub INDIRP, w których podane wektory prawidłowo wskażą powierzchnię.
 - Należy sprawdzić, czy istnieje prawidłowe ustawienie narzędzia względem powierzchni.
1201. Niewłaściwy rodzaj powierzchni (ANRMAI). Należy się porozumieć z programistą systemowym.
1401. Niewłaściwy rodzaj powierzchni (ADDST). Należy się porozumieć z programistą systemowym.
1402. Powierzchnia prostokreślna próbuje odwołać się do niewłaściwego rodzaju powierzchni (ADDST). Należy się porozumieć z programistą systemowym.
1404. Nieprawidłowa postać kanoniczna w zbiorze PROTAP. Należy się porozumieć z programistą systemowym.
- 1601. Ostrzeżenie - narzędzie znajduje się po niewłaściwej stronie powierzchni prowadzącej lub błędnie określono kierunek.
2101. Wystąpił błąd wejścia/wyjścia podczas zapisu zbioru danych CLTAPE (ATAPE). Należy się porozumieć z programistą systemowym.
2102. Wystąpił błąd wejścia/wyjścia podczas zapisu zbioru danych CLTAPE (ATAPE).
2201. Przekroczono dopuszczalną liczbę komunikatów dla jednego ciągu wektorów drogi narzędzia (AERR).
2202. Możliwość wystąpienia pętli w jednostce arytmetycznej. Należy się porozumieć z programistą systemowym.
3501. Punkty opisujące wybranie ułożeniowe powtarzają się (APOCK).
3502. Część punktów opisujących wybranie ułożeniowe jest współliniowa (APOCK).
3503. Punkty opisujące wybranie ułożeniowe nie leżą w jednej płaszczyźnie (APOCK).
3504. Punkty opisujące wybranie ułożeniowe tworzą figurę wklęsłą (APOCK).
3505. Wystąpił błąd wejścia/wyjścia podczas zapisu zbioru danych POCTAP (APOCK). Należy się porozumieć z programistą systemowym.
3506. Wystąpił błąd wejścia/wyjścia podczas odczytu zbioru danych POCTAP (APOCK). Należy się porozumieć z programistą systemowym.
3507. Kąt między osią narzędzia a płaszczyzną dna wybrania ułożeniowego jest za mały (mniejszy od 12°) lub parametr PTYPE ma wartość ujemną (APOCK). Jeżeli błąd powtarza się, należy się porozumieć z programistą systemowym.

3508. Wystąpił błąd wejścia/wyjścia podczas zapisu zbioru danych CLTAPE ze zbioru APOCK (APOCK).
Należy się porozumieć z programistą systemowym.
3509. Wystąpił błąd wejścia/wyjścia podczas przewijania do tyłu zbioru danych POCTAP (APOCK).
Należy się porozumieć z programistą systemowym.
3510. Podany w zbiorze APOCK promień narzędzia ma wartość ujemną lub zerową (APOCK).
3511. Podany w instrukcji POCKET efektywny promień narzędzia ma wartość ujemną.
3512. Podane w instrukcji POCKET odsunięcie ma wartość ujemną.
3513. Podane w instrukcji POCKET odsunięcie przy przejściu wykańczającym ma wartość ujemną.
3514. Jedna lub więcej z szybkości posuwu podanych w instrukcji POCKET ma wartość ujemną.
3515. Wskaźnik "0" w instrukcji POCKET nie przyjmuje jednej z wartości 0,1,2 lub 3.
3516. Wskaźnik rodzaju punktów w instrukcji POCKET nie jest równy 0 lub 1.
3517. Wystąpił błąd podczas testowania parametrów instrukcji POCKET.
3518. Narzędzie nie będzie odpowiednie do wykonania wybrania ulzeniowego.
3519. Powierzchnia przedmiotu dla wybrania ulzeniowego nie jest płaszczyzną.
- 3520. Ostrzeżenie - podano instrukcję AUTOPS w sytuacji, gdy oś narzędzia nie jest wektorem (0,0,1) oraz zostały podane modyfikatory TO,PAST lub obowiązuje modyfikator TLOFPS dla ustawienia względem powierzchni przedmiotu (AREPRE).

D O D A T E K D

Komunikaty o błędach wykrywanych w fazie edycji

Poniżej podany będzie spis błędów wykrywanych w fazie edycji. Błędy o numerach od 15 do 39 są związane z wykonywaniem instrukcji VTLAXS lub WCORN i na listingu (drukowanym w fazie edycji) podawane są w następującej postaci:

***** SECTION 3 5AXIS ERROR NUMBER ... *****

gdzie w miejscu oznaczonym jako ... podawany jest odpowiedni numer błędu.

1. Błąd taśmy magnetycznej w module SECTN3 na jednostce xxxx.
2. Należy sprawdzić wydruk CLTAPE z powodu błędu w programie.
3. Został wykryty koniec NUCLTP przed zapisem tego rekordu.
4. Nastąpiło przepełnienie tablicy indeksów. W programie obróbki części wystąpiło za dużo instrukcji INDEX/n.
5. Wystąpiło za dużo poziomów kopiowania (maksymalna dopuszczalna liczba poziomów kopiowania wynosi 3).
6. Niepoprawny modyfikator w instrukcji COPY.
7. Liczba kopiowań przekracza podaną liczbę n; należy sprawdzić program obróbki części, poszukując niedozwolonego przejścia do pętli kopiowania.
8. Niepoprawna liczba parametrów w instrukcji COPY na karcie numer xxx. Ostatnio pobrano kartę nr xxx.
9. Liczba n w instrukcji COPY na karcie nr xxx jest ujemna. Ostatnio pobrano kartę nr xxx.
10. Odbicie zwierciadlane nie jest dozwolone.
11. Błąd logiczny w instrukcji kopiowania.

12. Błędny rzut lub jego brak w instrukcji kreślenia ~~xxx~~. Karta nr ~~xxx~~.
13. Dolny narożnik ramki kreślącej jest algebricznie większy niż górny narożnik. Ramka została usunięta. Karta nr ~~xxx~~.
14. Niedozwolony modyfikator w parametrach postprocesora - karta nr ~~xxx~~, rekord nr ~~xxx~~.
15. Podana była instrukcja ~~VTLAXS/OFF~~ bez uprzedniego podania instrukcji VTLAXS/ON.
16. Występują dwie instrukcje ~~VTLAX/OFF~~ bez instrukcji VTLAXS/OFF między nimi.
17. Instrukcja FROM wystąpiła pomiędzy parą instrukcji VTLAXS/ON i VTLAXS/OFF.
18. Przed instrukcją VTLAXS/ON nie wystąpiła żadna instrukcja ruchu ani instrukcja FROM.
19. Nie użyto instrukcji MULTAX.
20. Określony był kąt nachylenia narzędzia większy niż 80° .
21. Promień narożny narzędzia jest większy niż podany promień narzędzia.
22. Wystąpił modyfikator ZSMALL w sytuacji, gdy narożny promień narzędzia nie jest zerowy.
23. Nie wystąpiła żadna instrukcja ruchu między instrukcją VTLAXS/ON a odpowiadającą jej instrukcją VTLAXS/OFF.
24. Więcej niż jedna instrukcja WCORN była użyta między parą instrukcji VTLAXS/ON i VTLAXS/OFF.
25. Kąt narożnika, który miał być wykonany, jest większy niż 180° .
26. Nie można znaleźć początkowego położenia narzędzia dla przejścia przy wykonywaniu narożnika. Przed wykonaniem narożnika należy sprawdzić odkształcenie krzywizny powierzchni opisane za pomocą drogi narzędzia po powierzchni przedmiotu. Należy również sprawdzić parametry w instrukcjach VTLAXS i WCORN.
27. Nie znaleziono zadowalającego położenia początkowego i końcowego dla przejścia narzędzia przy wykonywaniu narożnika. Należy sprawdzić odkształcenie krzywizny powierzchni, opisane za pomocą drogi narzędzia po powierzchni przedmiotu wokół narożnika, który miał być wykonywany. Należy też sprawdzić parametry w instrukcjach VTLAXS i WCORN.
28. Nie można znaleźć położenia końcowego dla przejścia narzędzia przy wykonywaniu narożnika. Należy sprawdzić odkształcenie krzywizny powierzchni, opisane za pomocą drogi narzędzia po powierzchni przedmiotu zaraz po wykonaniu narożnika. Należy również sprawdzić parametry w instrukcjach VTLAXS i WCORN.
29. Między instrukcjami VTLAXS/ON i WCORN określone było przesunięcie niewystarczające, aby rozpocząć proces wykonywania narożnika.
30. Określono niewystarczające przesunięcie między instrukcjami WCORN i VTLAXS/OFF, aby ukończyć proces wykonywania narożnika.
31. Instrukcja INDEX pojawiła się między instrukcją VTLAXS/ON i poprzedzającą ją instrukcją ruchu.
32. Instrukcja TRACUT była użyta między instrukcjami VTLAXS/ON i VTLAXS/OFF.
33. Instrukcja COPY została użyta między instrukcjami VTLAXS/ON i VTLAXS/OFF.
34. Pojawiło się niedozwolone żądanie wejścia/wyjścia lub wykryto EOF na nowym zbiorze CLTAPE.
35. Pojawił się błąd wejścia/wyjścia podczas odczytu starego zbioru CLTAPE.
36. Wystąpił błąd wejścia/wyjścia podczas zapisu nowego zbioru CLTAPE.
37. Wykryto błąd przy normalizacji wektora.
38. Pojawiło się niedozwolone żądanie wejścia/wyjścia lub wykryto koniec zbioru na starym zbiorze CLTAPE.

39. Pomiędzy instrukcjami VTLAXS/ON i VTLAXS/OFF wystąpiła instrukcja INDEX.

D O D A T E K E

Suplement

W części I niniejszego opracowania (BIOSK 1979 nr 5-6) opuszczony został punkt 5.18 o poniższej treści.

5.18. Dodatkowe typy elementów geometrycznych

W pewnych instalacjach systemu APT mogą być potrzebne dodatkowe typy elementów geometrycznych. W tym celu do zbioru słów kluczowych języka APT zostały dołączone następujące słowa:

CONSEC

LOFT

MESH

TOOL

Słowom tym nie zostało przypisane żadne znaczenie.

Na zakończenie cyklu na temat programowania
w języku APT podajemy spis treści poprzed-
nich części opracowania mgr inż. Haliny
Gutowskiej "Oprogramowanie w języku APT"

(Cz.I - współautorstwo mgr inż. Grażyny Kwaśniewskiej)

Cz. I - opublikowana w Biuletynie Informacyjnym
OSK 5-6 1979

	str.
1. Wstęp	4
2. Pojęcia podstawowe	4
2.1. Zbiór znaków dopuszczalnych w języku APT	5
2.2. Elementy języka APT	5
2.3. Instrukcje języka APT	10
2.4. Konwencje zapisu instrukcji	10
3. Zmienne i elementy geometryczne	11
3.1. Deklaracja tablicy	12
3.2. Lista indeksów	12
4. Instrukcja podstawienia arytmetycznego	14
4.1. Wyrażenie arytmetyczne	14
4.2. Funkcje standardowe	15
5. Instrukcje definicji geometrycznych	17
5.1. Definicja punktu	19
5.2. Definicja rozkładu punktów	26
5.3. Definicja wektora	33
5.4. Definicja prostej	39
5.5. Definicja okręgu	48
5.6. Definicja elipsy	55
5.7. Definicja hiperboli	57
5.8. Definicje krzywych drugiego stopnia	57
5.9. Definicje krzywych czwartego stopnia	58
5.10. Definicje płaszczyzny	60
5.11. Definicja kuli	64
5.12. Definicja walca	67
5.13. Definicja stożka	67
5.14. Definicja powierzchni stopnia drugiego	68
5.15. Definicja walca tabelarycznego	70
5.16. Definicja powierzchni wielostokowej	74
5.17. Definicja powierzchni prostokątnej	76
6. Definicja układu współrzędnych	80
6.1. Instrukcja REFSYS	81
6.2. Definicje macierzy	84
7. Definicje zagnieżdżone	94
8. Postać kanoniczna definicji	94
8.1. Spis postaci kanonicznych elementów geometrycznych i macierzy	95
8.2. Instrukcja CANON	101
9. Przykłady definiowania kształtu części	104

Cz. II - opublikowana w Biuletynie Informacyjnym OSK nr 4 1981

10. Informacje wstępne o programowaniu drogi narzędzia	4
11. Definicja narzędzia	5
11.1. Podstawowa instrukcja definicji narzędzia	5
11.2. Specjalna postać definicji narzędzia	7
12. Definicje zakresu tolerancji	7
12.1. Proste instrukcje tolerancji.....	7
12.2. Rozszerzone instrukcje tolerancji	9
13. Programowanie punktowe	9
13.1. Instrukcja początkowego ustawienia narzędzia	9
13.2. Instrukcja ruchu bezwzględnego	11
13.3. Instrukcja ruchu przyrostowego	11
13.4. Instrukcje ruchu punktowego wykorzystujące rozkłady punktów	12
14. Powierzchnie wymagane przy opisie drogi narzędzia	17
14.1. Powierzchnia przedmiotu	18
14.2. Powierzchnia prowadząca	20
14.3. Powierzchnia ograniczająca	22
15. Przesuw wstępny narzędzia	23
15.1. Instrukcje INDIRV i INDIRP	24
15.2. Instrukcje przesuwu wstępnego	26
15.3. Instrukcja SREVCT	40
15.4. Instrukcja OFFSET	42
16. Instrukcje ruchu narzędzia	45
16.1. Modyfikatory kierunkowe	47
16.2. Niejawnie określona powierzchnia ograniczająca	53
16.3. Wielokrotna powierzchnia ograniczająca	54
16.4. Wielokrotne przecięcie powierzchni ograniczającej	56
16.5. Informacje dodatkowe o modyfikatorach pozycyjnych TO, ON, PAST, TANTO.....	57
16.6. Uwagi dodatkowe o możliwości popełnienia błędów związanego z ustawieniem narzędzia względem powierzchni...	62
17. Instrukcja THICK.....	67
18. Instrukcja POCKET	69
18.1. Droga narzędzia przy wykonywaniu wybrania ulżeniowego	72
18.2. Obliczenie odległości pomiędzy kolejnymi przejściami narzędzia wokół wybrania ulżeniowego.....	74
19. Przekształcanie drogi narzędzia	77
19.1. Instrukcja przekształcania drogi narzędzia - TRACUT	77
19.2. Instrukcja kopiowania drogi narzędzia - COPY.....	79
20. Instrukcje specjalne w języku APT	87
21. Przykłady prostych programów obróbki części	87

Warunki prenumeraty wydawnictw
Instytutu Maszyn Matematycznych

Prenumeratę na kraj przyjmują Oddziały RSW "Prasa-Książka-Ruch" oraz urzędy pocztowe i doręczyciele w terminie do dnia 25 listopada na rok następny.

Jednostki gospodarki społecznej, instytucje, organizacje i wszelkiego rodzaju zakłady pracy zamawiają prenumeratę w miejscowych Oddziałach RSW "Prasa-Książka-Ruch", w miejscowościach zaś, w których nie ma Oddziałów RSW - w urzędach pocztowych.

Czytelnicy indywidualni opłacają prenumeratę wyłącznie w urzędach pocztowych i u doręczycieli.

Prenumeratę ze zleceniem wysyłki za granicę przyjmuje RSW "Prasa-Książka-Ruch", Centrala Kolportażu Prasy i Wydawnictw, ul. Towarowa 28, 00-958 Warszawa, konto PKO nr 1153-201045.

Prenumerata ze zleceniem wysyłki za granicę jest droższa od prenumeraty krajowej o 50% dla zleciennodawców indywidualnych i o 100% dla zlecających instytucji.

Ceny prenumeraty rocznej:

Biuletyn Informacyjny - Nauki i Techniki Komputerowe	- 1560,-
Przegląd Dokumentacyjny - Nauki i Techniki Komputerowe	1260,-
Prace Naukowo-Badawcze IMM - cena jednego zeszytu	- 220,-
cena prenumeraty rocznej	660,-

Zamówienia na prenumeratę Prac Naukowo-Badawczych IMM przyjmuje Dział Sprzedaży Wysyłkowej Ośrodka Rozpowszechniania Wydawnictw Naukowych PAN, Warszawa Pałac Kultury i Nauki, tel. 20-02-11 w. 2516. Egzemplarze pojedyncze Prac Naukowo-Badawczych IMM są do nabycia w księgarni Ośrodka Rozpowszechniania Wydawnictw Naukowych PAN, Warszawa, Pałac Kultury i Nauki, tel. 20-02-11 w. 2105.

BIBLIOTEKA GŁÓWNA
Politechniki Śląskiej

P

3057/82