

INSTYTUT ORGANIZACJI I ZARZĄDZANIA
POLITECHNIKI WROCŁAWSKIEJ

METODA POSZUKIWANIA SEKWENCJI PRZEKSZTAŁCAJĄCEJ
W PRZESTRZENI STANÓW
DLA POTRZEB PLANOWANIA PROCESÓW PRODUKCYJNYCH

Rozprawa doktorska

Zbigniew Szpunar

Promotor:

Prof. dr hab. inż. Bronisław Pilawski

WROCŁAW 1984

SPIS TREŚCI

1. WSTĘP	1
2. UWAGI NA TLE DOSTĘPNEJ LITERATURY	5
2.1. Krótka historia problematyki sztucznej inteli- gencji	5
2.2. Poszukiwanie rozwiązań jako kierunek badań nad AI	8
2.3. System poszukiwania rozwiązań	11
3. SFORMUŁOWANIE PROBLEMU	17
3.1. Określenia podstawowe	17
3.2. Język opisu zadania (JOZ)	19
3.3. Użycie JOZ do przedstawiania środowiska, stanów, sytuacji oraz reguł przekształcania	20
4. OPIS PODSTAWOWYCH AKCJI, WYSTĘPUJĄCYCH W METODZIE ..	24
4.1. Tworzenie opisów nowych sytuacji	24
4.2. Rozkład celu na podcele	26
4.3. Analiza drzewa podcelów i budowa drzewa rozwią- zania	36
5. PROPOZYCJA METODY POSZUKIWANIA SEKWENCJI PRZEKSZTAŁ- CAJĄCEJ W PRZESTRZENI STANÓW	46
5.1. Opis wariantu A metody - strategia rozwoju "wszerz"	46
5.2. Opis wariantu B metody - strategia rozwoju "w głąb"	57
5.3. Opis wariantu C metody - strategia uporządkowa- nego wyboru	58
5.3.1. Funkcja oceny dla wierzchołków w drzewie podcelów	58
5.3.2. Funkcja oceny dla wierzchołków w drzewie rozwiązania	61
5.3.3. Opis kroków wariantu C metody dwukierun- kowego poszukiwania sekwencji przekształ- cającej w przestrzeni stanów, z użyciem strategii uporządkowanego wyboru	63

5.4. Automatyczna modyfikacja funkcji oceny jako mechanizm adaptacyjny metody	67
5.5. Uwagi na temat możliwych zastosowań metody	70
6. PROGRAMOWA REALIZACJA PROPONOWANEJ METODY	71
6.1. Informacje wstępne	71
6.2. Przeznaczenie i ogólne zasady korzystania z systemu poszukiwania rozwiązań	71
6.3. Język komunikacji użytkownika z systemem	73
6.3.1. Język komunikacji organizacyjnej	73
6.3.2. Język komunikacji merytorycznej	84
6.4. Organizacja procesu poszukiwania rozwiązania zadania problemowego	86
6.5. Organizacja programowa systemu SPR	89
6.6. Przykład użycia systemu SPR do rozwiązywania zadania	91
7. WNIOSKI	122
LITERATURA	126

1. WSTĘP

Tematyka rozprawy tkwi w dziedzinie sztucznej inteligencji. Idea przedstawionej tu metody poszukiwania sekwencji przekształcającej w przestrzeni stanów zrodziła się pod wpływem studiów literaturowych, głównie z zakresu automatycznego rozwiązywania problemów, z uwzględnieniem aktualnych w latach 1979 + 1981 tendencji rozwoju tej problematyki.

Działania na rzecz opracowania metody były poprzedzone pracami badawczymi autora w zespole, budującym systemy komputerowego wspomagania dydaktyki: DIALOS/MLK [4,29,33] w latach 1977 + 1979 oraz DIALOS/INT [5,32,34,35] w latach 1979 + 1980, w ramach Problemu Resortowego RI-14 MNSzWiT "Rozwój Komputeryzacji Szkół Wyższych". Szczególnie cenne okazały się doświadczenia, zebrane przez autora podczas projektowania, oprogramowania i eksperymentalnej eksploatacji systemu DIALOS/INT. Opracowana wówczas przez autora rozprawy i zastosowana w systemie DIALOS/INT metoda dwukierunkowego poszukiwania sekwencji przekształcającej [30,32] została wzbogacona i rozwinięta do postaci, przedstawionej w niniejszej rozprawie, a następnie oprogramowana w sposób, zapewniający kompatybilność z istniejącym już oprogramowaniem i bazą danych systemu DIALOS/INT.

Tezą niniejszej rozprawy jest wykazanie, że
(T) "istnieje metoda poszukiwania sekwencji przekształcającej w przestrzeni stanów, korzystająca z:

- języka opisu zadań, opartego na języku rachunku predyktów,
- dwukierunkowego poszukiwania, tzn. współzależnego, wieloetapowego rozbijania sytuacji docelowej na podcele składowe oraz wieloetapowego przekształcania sytuacji początkowej,
- zbioru funkcji heurystycznych dla określania perspektywicznych kierunków poszukiwania rozwiązania,

- mechanizmu zbieżności do rozwiązania quasi-optymalnego,
- mechanizmu automatycznej modyfikacji funkcji oceny,

umożliwiającego gromadzenie doświadczeń i korzystanie z nich w rozwiązywaniu analogicznych problemów, i umożliwiająca automatyczne agregowanie ogólnych reguł przekształcania w przestrzeni stanów w uporządkowany ciąg, przekształcający sytuację początkową w sytuację docelową".

Oryginalność metody, zdaniem autora rozprawy, polega na łącznym użyciu:

1^o specjalnego języka opisu zadań (problemów) dla metody, opartego na języku rachunku predykatów; język ten umożliwia przedstawianie zadań z różnych dziedzin w sposób zrozumiały dla metody (komputera), podnosząc uniwersalność jej zastosowań. Między innymi w proponowanym języku opisu zadań można opisywać środowisko problemowe zadania oraz przedstawiać tzw. reguły przekształcania w ogólnej, sparametryzowanej postaci, na różnych poziomach szczegółowości.

2^o strategii dwukierunkowego poszukiwania sekwencji przekształcającej; jej użycie umożliwia znaczne zawężenie zbioru analizowanych przez komputer wariantów oraz ukierunkowuje poszukiwania na osiągnięcie celu.

3^o zbioru funkcji heurystycznych, dla podniesienia efektywności procesu poszukiwania sekwencji przekształcającej, poprzez ocenę perspektywiczności różnych, alternatywnych kierunków poszukiwania rozwiązania i wybór kierunku, rokującego najwięcej szans na sukces.

4^o metody automatycznej modyfikacji funkcji oceny (na podstawie zdobywanych doświadczeń), prowadzącej do sukcesywnego doskonalenia strategii metody poszukiwania sekwencji przekształcającej (poprzez uczenie się na własnych błędach), co umożliwia zbliżanie się do rozwiązań quasi-optymalnych.

Przedstawieniu i rozwiązaniu postawionego problemu poświęcone są rozdziały 2 ÷ 7 niniejszej rozprawy.

W rozdziale 2. rozprawy omówione są wybrane techniki tzw. rozwiązywania problemów, występujące w problematyce sztucznej inteligencji i wiążące się z tematem rozprawy.

Prawdziwość tezy (T), poprzez podanie opisu metody oraz opisu oprogramowania, wykazano w niniejszej rozprawie w rozdziałach 3 ÷ 6.

Rozdział 3. formułuje problem i wprowadza język opisu zadań dla metody.

Rozdział 4. omawia podstawowe rodzaje akcji, występujących w metodzie.

Rozdział 5. zawiera opisy metody poszukiwania sekwencji przekształcającej w trzech wariantach, wynikających z użycia trzech różnych strategii ogólnych, przedstawia też funkcje ocen wraz ze sposobem ich użycia w metodzie. Prezentuje ponadto mechanizm zbieżności do rozwiązania quasi-optimalnego oraz mechanizm adaptacyjny metody.

Rozdział 6. zawiera opis programowej realizacji metody oraz przykłady użycia zbudowanego oprogramowania do rozwiązywania zadań.

Rozprawę kończą wnioski w rozdziale 7.

Programową realizację metody stanowi tzw. system poszukiwania rozwiązań SPR (opisany w rozdziale 6.), składający się z czterech modułów programowych, napisanych w języku programowania FORTRAN 1900 oraz pewnej liczby kartotek systemowych, tworzących bazę danych. System ten został uruchomiony i przeszedł eksploatację próbną w Centrum Obliczeniowym Politechniki Wrocławskiej, na komputerze ODRA 1305, pod kontrolą systemu operacyjnego GEORGE 3.

Autor rozprawy doktorskiej poczuwa się do miłego obowiązku podziękowania doc. dr Tadeuszowi Huskowskiemu za cenne wskazówki, udzielone podczas przygotowywania niniejszej rozprawy oraz Koleżankom i Kolegom z byłego zespołu badawczego DIALOS za ich współudział w pracach na rzecz systemu wspomaganie dydaktyki DIALOS/INT, stanowiącego element oprogramowania przedstawionej tu metody.

2. UWAGI NA TLE DOSTĘPNEJ LITERATURY

2.1. Krótką historia problematyki sztucznej inteligencji

Pod pojęciem "sztuczna inteligencja" (AI - od ang.: artificial intelligence) będziemy rozumieć takie zaprogramowanie komputera, aby zachowywał się on w sposób uznany powszechnie za inteligentny [48].

Rozwój badań nad zagadnieniami AI został zapoczątkowany w późnych latach pięćdziesiątych i postępuje w dwóch podstawowych kierunkach. Pierwszy z nich ma na celu odkrywanie, rozumienie i badanie metodami naukowymi elementarnych procesów, na których opiera się zachowanie inteligentne. Celem drugiego kierunku, związanego z zastosowaniami, jest budowanie programów komputerowych, które wykazują zachowanie inteligentne, działając w różnych dziedzinach praktycznych.

W rozwoju badań nad AI można wyodrębnić dwa okresy. Pierwszy okres, od późnych lat pięćdziesiątych do wczesnych lat siedemdziesiątych, został poświęcony głównie rozwojowi podstaw AI. Położono nacisk na ogólność metod i pojęć, względnie mało uwagi poświęcając sprawności otrzymywanych programów bądź systemów. Zbudowane systemy stosowano głównie w ściśle opisanych matematycznie dziedzinach, jak gry, dowodzenie twierdzeń, manipulowanie na symbolach i proste planowanie działań robotów. W okresie tym wyłoniły się podstawowe zagadnienia AI:

1. Rozwiązywanie problemów - jako dziedzina poszukiwania rozwiązań zadań (problemów), dla których nie jest znany z góry żaden prosty sposób rozwiązania;
2. Reprezentacja wiedzy, której celem jest przedstawianie faktów, praw, akcji lub sądów tak, aby mogły one być efektywnie użyte przez inne części systemu inteligentnego;
3. Postrzeganie, którego celem jest wybieranie potrzebnej in-

formacji z "zewnętrznego świata", np. rozpoznawanie obrazów, widzenie, pisany i mówiony język naturalny, itd.

Ogólne techniki, rozwinięte w tych podstawowych dziedzinach, obejmują między innymi: poszukiwanie heurystyczne [46], rozkład problemu na podproblemy [48], zasadę rezolucji [51] i inne mechanizmy wnioskowania w dowodzeniu twierdzeń, sieci semantyczne [49], proceduralne przedstawianie wiedzy, sieci proceduralne [53].

W drugim okresie rozwoju AI, od wczesnych lat siedemdziesiątych, głównym kierunkiem było tworzenie systemów stosujących AI do problemów, występujących w praktyce. Charakterystyczne dla tego kierunku były i są usiłowania połączenia kilku składników, należących do różnych dziedzin, w jeden system i zrozumienia, jak te składniki muszą być dopasowane funkcjonalnie, aby wytworzyć zachowanie inteligentne. Inny nurt kładł nacisk na systemy bazujące z jednej strony na szczegółowej wiedzy z danej dziedziny, z drugiej zaś na ogólnych technikach, by systemy te mogły wykazywać wysoką fachowość. Takie systemy działają dziś w różnych dziedzinach, wykazując niejednokrotnie bardzo dużą skuteczność, nawet w porównaniu ze skutecznością specjalistów ([2],[6],[53]).

Równolegle z rozwojem metod i systemów AI ma miejsce rozwój tzw. języków bardzo wysokiego poziomu, zorientowanych na pisanie programów heurystycznych. Powstają języki: PLANNER [24], PROLOG [52], CONNIVER [59], QA4 [7], QLISP [50] i inne, których udane implementacje są niezwykle pomocnym narzędziem dla projektanta-programisty, budującego system AI ([62],[66]). Obejmują one większość mechanizmów, które wystąpiły w programach heurystycznych i systemach ogólnych wcześniejszego okresu. Ich istotą jest opisywanie zadania i jego środowiska problemowego a nie określanie sposobu rozwiązywania tego zadania,

jak to ma miejsce w klasycznych językach programowania (Fortran, Algol, Lisp) [22].

Postępy w pracach nad AI, często koncentrujących się na styku wielu dyscyplin naukowych, znajdują bogate odzwierciedlenie w literaturze na ten temat. Dostępne są między innymi monografie na temat AI: Benerji'ego [1], Bundy'ego [3], Efi-mowa [11], Feigenbauma [14], Gładuna [19], Góralskiego [20, 21, 22, 23], Hożyńskiego [25], Hunta [26], Nilssona [48], Sacerdoti'ego [53,54], Slagle'a [57], Winstona [67] oraz monografie z dziedzin pokrewnych, silnie związanych z AI (między innymi automatycznego programowania ([40],[41]), teorii układów uczących się [14], automatycznego projektowania ([27],[28], [53]) i innych). Ponadto ukazuje się szereg czasopism o zasięgu międzynarodowym, w tym przede wszystkim wydawane w formie książki, traktujące wyłącznie o problemach AI: "Artificial Intelligence" (1 + 4 tomów rocznie, ukazuje się od 1970 roku nakładem wydawnictwa North-Holland) oraz "Machine Intelligence" (1 + 2 tomów rocznie, ukazuje się od 1969 roku, nakładem wydawnictwa Uniwersytetu w Edynburgu). Dostępne są również materiały konferencyjne (wydawnictwa North-Holland) z międzynarodowych konferencji na temat AI "International Joint Conference on Artificial Intelligence", odbywających się od 1969 roku co dwa lata w dużych ośrodkach uniwersyteckich.

Na czwartej z kolei konferencji (Tbilisi 1975r) wprowadzono następującą klasyfikację problemów w ramach AI [25]:

1. Matematyczne i teoretyczne aspekty AI,
2. Reprezentacja problemów i wiedzy,
3. Planowanie i rozwiązywanie problemów,
4. Uczenie, korekcja błędów i automatyczne programowanie,
5. Technika poszukiwania,
6. Rozumienie języka naturalnego,

7. Rozumienie mowy,
8. Oprogramowanie AI,
9. Przetwarzanie informacji wizualnej,
10. Roboty i technologie produkcyjne,
11. Różne zastosowania AI,
12. Psychologiczne aspekty AI.

Tematyka niniejszej rozprawy tkwi głównie w zakresie planowania i rozwiązywania problemów, technik poszukiwania rozwiązań oraz oprogramowania AI.

2.2. Poszukiwanie rozwiązań jako kierunek badań nad AI

W literaturze poświęconej AI termin "rozwiązywanie zadań" ma ściśle określony sens. Z zadaniem mamy do czynienia wówczas, gdy opisane są:

- stan początkowy (A) pewnego obiektu lub zbiór stanów początkowych pewnych obiektów (tzw. sytuacja początkowa),
- charakterystyczne elementy sytuacji docelowej (B),
- operacje (przekształcenia), za pomocą których można przejść od jednego stanu do drugiego (tzw. operatory lub reguły przekształcania).

Dodatkowo mogą wystąpić pewne ograniczenia, np. w postaci wykazu stanów zabronionych.

Takie określenie zadania, dokonane w oparciu o przestrzeń stanów, jest najczęściej spotykane w literaturze ([26],[48],[67]).

Ścisła definicja zadania dla metody, opisywanej w niniejszej rozprawie, będzie podana w p. 3.1.

Rozwiązaniem zadania ma być sekwencja operacji (przekształceń), zapewniająca przejście od A do B. Uzyskiwanie tej sekwencji metodą sprawdzania wszystkich możliwych kombinacji na ogół jest niewykonalne, ze względu na mnogość alternatywnych

wariantów, które należy przeanalizować. Problem nie znika również wtedy, gdy rolę organizatora przeglądu przejmuje komputer. Takie postępowanie jest co prawda systematyczne, lecz nieukierunkowane i przez to nieskuteczne.

Przeciwnieństwem takiego "algorytmicznego" podejścia jest podejście heurystyczne *) (termin ten wprowadził Polya w 1954 roku, bez podania jego formalnej definicji), zastosowane przez grupę: Newell, Shaw i Simon [43], oparte na obserwacji sposobu dowodzenia twierdzeń w matematyce. Szczególnie ważne okazało się ukierunkowywanie wyводу, dokonywane na podstawie jego bieżącej analizy oraz poprzez porównywanie z osiąganym stanem finalnym [26]. W roku 1957 tworzą oni program LOGIK-TEORETYK [43], przeznaczony do dowodzenia twierdzeń, uznawany za pierwszy program, realizujący automatyczne poszukiwanie rozwiązania. Ogólna organizacja poszukiwania rozwiązania, zastosowana w tym programie, daje początek serii programów, między innymi: dowodzenia twierdzeń geometrycznych [18], całkowania symbolicznego (program Slagle'a SIMON [56] z 1963 roku i jego kolejna, udoskonalona wersja z 1967 roku, program MULTIPLE [58] z 1968 roku), układania twierdzeń opartych na zadanych aksjomatach (Uniwersytet Berkeley - 1967 rok), czy wreszcie samouczenia się w programie analizy regresyjnej [42] (Massachusetts Institute of Technology).

Ogromny wpływ na kierunki i poziom dalszych badań w dziedzinie automatycznego poszukiwania rozwiązań miały prace grupy: Newell, Shaw, Simon i Ernst, związane z programem GPS (General Problem Solver - Rozwiązywacz Problemu Ogólnego) ([12],[14]). Cechą szczególną GPS było użycie w poszukiwaniu

*) Autor, zdając sobie sprawę z wieloznaczności określenia "heurystyczny", ogranicza się do metodologicznego sensu tego pojęcia.

rozwiązań odpowiednich technik analizy celów i środków, umożliwiających tzw. planowanie rozwiązania. GPS, pozostając zgodnie z założeniami autorów programem ogólnym i uniwersalnym, miał naśladować stosunkowo wiernie sposób działania człowieka rozwiązującego zadanie. W odróżnieniu od programu LOGIK-TEORETYK, "osadzonego" w formalizmach klasy zadań, dla których był zbudowany, GPS dopuszczał możliwość użycia reguł przekształcania na abstrakcyjnym poziomie. Wybierając odpowiednie reguły przekształcania (dla zmniejszania "różnicy" między stanem bieżącym a docelowym) sprowadzał rozwiązanie zadania do rozwiązania zbioru zadań składowych, uzyskanych w wyniku dokonywanych sukcesywnie przekształceń stanów. Dzięki organizacji zapewniającej uniwersalność znacznie poszerzył obszar możliwych zastosowań automatycznego rozwiązywania. Użyty do rozwiązania dziesięciu nieskomplikowanych wprowadzie zadań, lecz tkwiących w różnych dyscyplinach (od całkowania symbolicznego do łamigłówek), w zestawieniu z rezultatami rozwiązań tych samych zadań przez ludzi, odniósł niekwestionowany sukces [26].

Równocześnie z GPS powstaje w Edynburgu program GRAPH TRAVERSER Dorana i Michie [8], który jako pierwszy korzystał w poszukiwaniu rozwiązania z funkcji oceny perspektywiczności danego kierunku rozwiązania.

W początkach lat siedemdziesiątych powstaje szereg monografii i ciekawych prac uogólniających rezultaty pierwszego okresu badań nad poszukiwaniem rozwiązań. Wśród nich szczególną pozycję zyskały sobie prace Nilssona ([46],[47],[48]), Slagle'a [57], Benerji'ego, Pospiełowa, Głuszkowa, Amosowa, Simona [45] i Sandwell'a. Wynika z nich między innymi, że system rozwiązywania problemów powinien mieć następujące cechy:

1. Posiadać podlegający modyfikacjom model środowiska proble-

mowego,

2. Wykazywać skuteczność w zastosowaniu do szerokiej klasy zadań,
3. Posiadać umiejętność planowania działań,
4. Posiadać umiejętność gromadzenia doświadczeń i uczenia się.

Skuteczne połączenie odpowiednich mechanizmów (metod), zapewniających spełnienie wymienionych funkcji w jednym systemie poszukiwania rozwiązań jest zagadnieniem, które w chwili obecnej wyznacza ważny kierunek badań nad sztuczną inteligencją.

2.3. System poszukiwania rozwiązań

Ogólną funkcją systemu poszukiwania rozwiązań jest konstruowanie sekwencji przekształceń (lub inaczej: akcji), transformujących pewien stan (lub zbiór stanów, zwany sytuacją) w przestrzeni stanów w inny stan (lub zbiór stanów) w tej przestrzeni. Dla spełnienia tego zadania system powinien być wyposażony w pewne mechanizmy warunkujące efektywne działanie systemu [15]. Są to:

- (1) Mechanizm zarządzania modelami opisów stanów.

Model opisu stanów jest specyfikacją stanów obiektów w przestrzeni. Zawarte są w nim fakty lub relacje, które zachodzą między obiektami. Mogą one być przedstawiane np. jako formuły zdaniowe języka rachunku predykatów [37]. Ważnym aspektem takiej reprezentacji przestrzeni stanów, z punktu widzenia poszukiwania rozwiązań, jest również konieczność przedstawiania wielu alternatywnych sekwencji przekształceń (tzw. przestrzeni poszukiwania rozwiązania), pojawiających się w modelu podczas przebiegu procesu poszukiwania rozwiązania. Istnieje szereg sposobów przedstawiania przestrzeni poszukiwania rozwiązania.

Na ogół stosowane są przedstawienia dostosowane do struktury sterowania, występującej w danym rozwiązywaczu problemów.

(2) Mechanizm dedukcji.

Model opisu stanów zawiera wszystkie informacje niezbędne do scharakteryzowania każdego pojedynczego stanu przestrzeni, lecz nie wszystkie informacje dane są wprost. Mechanizm dedukcji powinien zapewnić dostarczanie informacji potrzebnej w danym etapie procesu poszukiwania rozwiązania, poprzez pozyskiwanie jej z modelu opisów stanów (w tym ujęciu mechanizm dedukcji można rozpatrywać w uproszczeniu jako system "pytanie-odpowiedź").

(3) Modele przekształceń (akcji).

Rozwiązywacz problemów musi mieć możliwość przedstawiania zmian, wprowadzanych w modelu opisów stanów przez przekształcenia stanów, zachodzące w procesie poszukiwania rozwiązania. W związku z tym każde możliwe przekształcenie przedstawiane jest jako schemat (nazywany regułą przekształcania lub operatorem) przejścia od jednego stanu (lub zbioru stanów) do innego stanu (lub zbioru stanów). Często ma on postać procedury z parametrami, może też być przedstawiany za pomocą deklaratywnych struktur danych. W swym opisie reguła przekształcania zawiera najczęściej:

- wykaz warunków, które muszą być spełnione, by przekształcenie mogło mieć miejsce; rolę warunków pełnią opisy stanów, których istnienie warunkuje użycie reguły,
- wykaz stanów, które dane przekształcenie usuwa z przestrzeni poszukiwania rozwiązania,
- wykaz stanów, które dane przekształcenie wprowadza do przestrzeni poszukiwania rozwiązania.

Model przekształcenia może zawierać w swym opisie, w miejsce stanów, pewne zmienne, pełniące rolę argumentów danej re-

guły przekształcania. Wówczas ta reguła jest ogólnym schematem przekształcania rodziny stanów w inne stany lub rodziny stanów.

Konstruowanie sekwencji przekształcającej na ogół polega na tworzeniu i rozpatrywaniu pewnych alternatywnych przekształceń lub ciągów przekształceń w przestrzeni poszukiwania rozwiązania. Istniejące systemy poszukiwania rozwiązań z reguły pracują wstecz, od danego stanu docelowego, by znaleźć sekwencję przekształceń, która doprowadzi do tego stanu od stanu początkowego. Takie działanie prowadzi do utworzenia wielu wariantów przekształceń, dających się przedstawić graficznie w postaci drzewa, ze stanem docelowym jako korzeniem, regułami przekształcania jako gałęziami i stanami pośrednimi jako wierzchołkami. Proces rozbudowywania i analizowania drzewa prowadzony jest w celu znalezienia ścieżki do wierzchołka, odpowiadającego stanowi początkowemu. Ścieżka taka, od wierzchołka początkowego do wierzchołka końcowego, jest przedstawieniem poszukiwanej sekwencji przekształcającej.



Istnieje szereg strategii sterowania rozwojem przestrzeni poszukiwania rozwiązania, wywodzących się z dwóch grup strategii. Pierwszą z nich, zastosowaną między innymi w systemach GPS [44] i STRIPS [16], można w uproszczeniu przedstawić następująco:

1. Określ różnicę między stanami: początkowym a końcowym i wybierz regułę przekształcania, która maksymalnie redukuje tę różnicę.
2. Jeśli wybrana reguła może służyć do przekształcenia stanu początkowego, to po jej użyciu powstają nowe, pośrednie stany. Jeśli cel główny jest wśród nich - koniec, jest rozwiązanie w postaci wykonanego właśnie przekształcenia. W przeciwnym wypadku określana jest różnica między nowym

stanem a celem głównym i wybierana jest reguła, która maksymalnie ją redukuje, itd.

3. Jeśli wybrana reguła nie może być użyta, występujące w warunkach użycia reguły opisy stanów stają się nowymi celami do osiągnięcia i są nazywane, dla odróżnienia od celu głównego, podcelami. Stosując rekursywnie tę procedurę można znaleźć sekwencję przekształceń, osiagającą te podcele. Wówczas wybrana poprzednio reguła przekształcania może być użyta i poszukiwanie rozwiązania jest kontynuowane, aż do znalezienia rozwiązania lub do osiągnięcia założonych ograniczeń - warunków zaniechania procesu. Jeśli podcel nie może być osiągnięty, wybiera się kolejną regułę przekształcania dla zminimalizowania różnicy (jw.) i proces powtarza się.

Druga ważna strategia polega na zmienianiu kierunku poszukiwania rozwiązania. Nazywana jest ona niekiedy wycofywaniem się lub powrotem (od ang.: backtracking). Istotę tej strategii można w uproszczeniu przedstawić następująco:

1. Jeśli cel jest zawarty w sytuacji początkowej, wówczas znaleziono trywialne rozwiązanie. W przeciwnym wypadku wybierana jest reguła przekształcania, która umożliwia osiągnięcie celu głównego. Jeśli regułę tę można użyć do sytuacji początkowej, stosuje się ją i koniec, jest rozwiązanie w postaci wykonanego właśnie przekształcenia.
2. W przeciwnym wypadku stosuje się kolejną regułę umożliwiającą osiągnięcie stanów, występujących w warunkach użycia poprzednio rozpatrywanej reguły, itd.
3. Jeśli poszukiwanie zawiedzie (tzn. nie doprowadzi do rozwiązania po wykonaniu określonej liczby kroków jw.), wybiera się następną regułę przekształcania i proces powtarza się.

W tej strategii może dojść do analizy wszystkich wariantów przekształceń, co grozi znacznym wydłużeniem czasu oczekiwania na rozwiązanie. Mamy tu jednak do czynienia z możliwością efektywnego przedstawienia drzewa rozwiązania w postaci stosu, co znacznie upraszcza strukturę pamięci oraz korzystanie z informacji, uczestniczącej w procesie.

Przedstawione strategie ogólne są w praktyce niewystarczające dla poszukiwania rozwiązań wielu złożonych problemów. W szczególności, niezwykle kosztowne jest poszukiwanie rozwiązań zadań, w których cel główny wyrażony jest jako koniunkcja celów, bądź gdy podczas rozbijania celu powstaje wiele tzw. podcelów koniunkcyjnych.

Dla podniesienia efektywności praktycznych systemów poszukiwania rozwiązań rozwinięto w latach siedemdziesiątych wiele szczegółowych technik poszukiwania rozwiązań, nazywanych też taktykami rozwiązywania problemów [55]. Do ważniejszych taktyk można zaliczyć:

- planowanie hierarchiczne [54] - w chwili obecnej taktyka dominująca w systemach poszukiwania rozwiązań,
- dowodzenie nie wprost [65],
- pseudo-redukcja [62],
- wsteczna transformacja celów [64],[65],
- trafne wycofywanie się [38].

Lektura dostępnej literatury na temat poszukiwania rozwiązań pozwala sądzić, że przyszłość w tej dziedzinie należy będzie do systemów, godzących dużą elastyczność i uniwersalność z wysoką efektywnością procesu poszukiwania rozwiązania. Będzie to możliwe w systemach integrujących szereg taktyk, ze szczególnym uwzględnieniem taktyk sprzyjających zdobywaniu przez system doświadczeń w celu uczenia się rozwiązywania problemów podobnych. Trybem pracy takich systemów powin-

na być interakcja, umożliwiająca zespołowi człowiek-system sprzęgnięcie dużej mocy obliczeniowej komputera z doświadczeniem i intuicją użytkownika systemu.

Opisana w dalszych rozdziałach niniejszej rozprawy metoda, wraz z jej programową realizacją, uwzględniając realia i potrzeby rozwijającej się dynamicznie dziedziny poszukiwania rozwiązań, jest propozycją rozwiązania wybranych jej problemów.

3. SFORMUŁOWANIE PROBLEMU

3.1. Określenia podstawowe

Będziemy rozpatrywali następujące zbiory:

- a) Zbiór $A = \{a_1, a_2, \dots, a_n\}$, którego elementy będziemy nazywali obiektami; zbiór A będziemy nazywali środowiskiem.
- b) Zbiór $B = \{b_1, b_2, \dots, b_m\}$, którego elementy będziemy nazywali stanami obiektów. Będziemy niekiedy dodatkowo przyjmowali, że:
 - każdemu stanowi $b_k \in B$ jest przyporządkowany zbiór $\{\beta_1^k, \beta_2^k, \dots, \beta_p^k\}$, którego elementy będziemy nazywali składowymi stanu b_k ,
 - składowe stanu identyfikują stan jednoznacznie.

Definicja 1. Przestrzenią stanów będziemy nazywali relację $\mathcal{P} \subset A \times B$, której dziedziną jest zbiór A . Jeśli $(a_i, b_k) \in \mathcal{P}$, to będziemy mówili, że b_k jest stanem dopuszczalnym obiektu a_i , lub krócej: stanem obiektu a_i .

Definicja 2. Sytuacją będziemy nazywali podzbiór S przestrzeni stanów $\mathcal{P}$, będący przekrojem $\mathcal{P}$ nad A (tzn. przeciwdziedzinę funkcji określonej na zbiorze obiektów A i przyporządkowującej każdemu obiektowi $a_i \in A$ parę $(a_i, b_k) \in \mathcal{P}$). Sytuacja jest więc zbiorem par: (obiekt, stan).

W przestrzeni stanów $\mathcal{P}$ będziemy rozpatrywali przekształcenia $O_1: S_i \xrightarrow{O_1} S_j$, pozwalające z sytuacji S_i tworzyć sytuację S_j . Każde z rozpatrywanych przekształceń musi być określone za pomocą reguły przekształcania (oznaczonej tak samo jak przekształcenie).

Definicja 3. Regułą przekształcania $O_1 = \langle Co, Ca, Ad, Re \rangle$ nazywamy trzy zbiory Co, Ca i Ad par (obiekt, stan) oraz zbiór Re związków między elementami zbiorów Co, Ca i Ad . Za pomocą O_1 określamy sytuację S_j , gdy dana jest sytuacja S_i :

$$Co \subset S_i \Rightarrow S_j = S_i \setminus Ca \cup Ad$$

Z powyższej implikacji wynika rola zbiorów Co, Ca i Ad. Zbiór Co (ang.: conditions) zawiera pary (obiekt, stan), których występowanie w sytuacji S_i umożliwia utworzenie sytuacji S_j za pomocą O_1 . Zbiór Ca (ang.: cancelations) zawiera pary (obiekt, stan), na które oddziałują i które zmienia reguła O_1 . Zbiór Ad (ang.: additions) zawiera pary (obiekt, stan), które wprowadza reguła O_1 .

Jeśli $Co \not\subseteq S_i$, to przekształcenie O_1 dla sytuacji S_i nie jest określone. Zbiory Co, Ca i Ad mogą zawierać w ogólnym przypadku pewną ilość zmiennych q_i , $i=1,2,\dots$ w miejsce obiektów lub stanów. Zmienne te pełnią wówczas rolę argumentów reguły przekształcania a zbiory Co, Ca i Ad są rodzinami zbiorów. Uporządkowany zbiór argumentów reguły O_1 oznaczamy przez $Q_1 = (q_1, q_2, \dots, q_k)$. Zbiór Re jest zbiorem ("wykazem") funkcji wiążących argumenty reguły O_1 . Zbiory Co i Ad nie mogą być zbiorami pustymi, natomiast Ca i Re mogą być zbiorami pustymi.

Przedstawiona w rozprawie metoda ma służyć do automatycznego poszukiwania sekwencji przekształcającej $\mathcal{T}$

$$S_p \longrightarrow \dots \longrightarrow S_i \longrightarrow S_j \longrightarrow \dots \longrightarrow S_d$$

w przestrzeni stanów $\mathcal{P}$, na podstawie danych:

- sytuacji S_p zwanej początkową,
- podzbioru S_k sytuacji S_d zwanej docelową ($S_k \not\subseteq S_p$),
- zbioru reguł przekształcania $O = \{O_1, O_2, \dots, O_1, \dots, O_z\}$.

Z określenia tego wynika, że metoda jest dostosowana do zadań, gdy określona jest pewna klasa sytuacji, z których każda może być sytuacją docelową i przy tym każda z nich zawiera istotny dla problemu podzbiór S_k . Podzbiór S_k nazywać będziemy celem głównym.

Zadanie, które ma być rozwiązane przedstawioną metodą,

określa się tylko na podstawie:

- środowiska A ,
- tych spośród stanów zbioru B , które służą do przedstawienia sytuacji S_p oraz S_k ,
- zbioru reguł przekształcania O .

Użycie reguł ze zbioru O do przekształcania sytuacji umożliwi określenie tych sytuacji, które wystąpią w trakcie rozwiązywania w przestrzeni $\mathcal{P}$.

3.2. Język opisu zadania (JOZ)

Na użytek przedstawionej metody wprowadzono prosty język opisu zadania (JOZ), służący do opisywania stanów, sytuacji, środowiska i reguł przekształcania. Język ten jest oparty na języku rachunku predykatów [37],[64]. Podstawowymi elementami języka są:

- 1° Zmienna, reprezentująca obiekty należące do ustalonego zbioru. Typ zbioru określa typ zmiennej.
- 2° Stała, reprezentująca konkretny obiekt w danym zbiorze.
- 3° Predykat (czyli funkcja zdaniowa), której argumentami mogą być:
 - zmienne,
 - stałe.

Predykat przyjmuje wartość logiczną PRAWDA, gdy między jego argumentami zachodzi określony związek, lub wartość logiczną FAŁSZ w przeciwnym wypadku.

Podstawowe elementy JOZ służą do budowy złożonych konstrukcji, tzw. formuł. Formułą atomową jest predykat wraz z jego argumentami. Poprawnie zbudowaną formułą JOZ (lub krótko: formułą) może być:

- formuła atomowa,
- negacja formuły,

- alternatywa formuł,
- koniunkcja formuł,
- równoważność formuł,
- formuła poprzedzona kwantyfikatorem i symbolem zmiennej związanej.

3.3. Użycie JOZ do przedstawiania środowiska, stanów, sytuacji oraz reguł przekształcania

Poszczególne obiekty środowiska A są przedstawiane za pomocą stałych. Klasy obiektów reprezentowane są przez zmienne.

Stan obiektu środowiska A jest przedstawiany za pomocą jednej formuły f lub - w przypadku, gdy posługujemy się składowymi stanów - za pomocą zbioru formuł. Tym samym sytuacja S jest przedstawiana za pomocą zbioru formuł, który będziemy oznaczali również literą S :

$$S = \{f_1, f_2, \dots, f_n\}.$$

Podobnie przedstawiane są reguły przekształcania:

$$O_1 = \langle Co, Ca, Ad, Re \rangle.$$

Poszczególne zbiory Co , Ca , Ad i Re reguły O_1 przedstawiane są za pomocą zbiorów formuł; zbiory te będziemy oznaczali również przez Co , Ca , Ad i Re :

$$Co = \{co_1, co_2, \dots, co_p\},$$

$$Ca = \{ca_1, ca_2, \dots, ca_r\},$$

$$Ad = \{ad_1, ad_2, \dots, ad_s\},$$

$$Re = \{re_1, re_2, \dots, re_t\}.$$

W formułach opisujących zbiory: Co , Ca , Ad i Re mogą wystąpić zmienne $q_1, q_2, \dots$ (o nazwach zastrzeżonych), które będą pełnić rolę argumentów poszczególnych reguł przekształcania. Argumenty te mogą wystąpić w formułach w miejsce obiektów.

tów lub stanów.

Przy formułowaniu zadania w JOZ należy dążyć do prostoty i jednoznaczności opisów poszczególnych elementów zadania oraz unikać nadmiaru informacji w opisach reguł przekształcania. Wygodne jest korzystanie z formuł atomowych dla opisywania elementów przestrzeni stanów, występujących w sytuacji S_p , w podzbiorze S_k sytuacji S_d oraz w zbiorach Co, Ca, Ad i Re reguł przekształcania.

W dalszej części rozprawy, dla uproszczenia opisu przyjęto, że jeden element przestrzeni stanów $\mathcal{P}$, czyli jedna para (obiekt, stan), jest opisywany za pomocą jednej formuły.

Dla zilustrowania powyższych opisów zamieszczono prosty przykład zadania. Zadanie to będzie wykorzystywane również w dalszej części rozprawy, do ilustracji poszczególnych mechanizmów metody.

PRZYKŁAD. Środowisko A składa się z następujących obiektów: pokój, robot, skrzynka, wyłącznik. Robot, skrzynka i wyłącznik znajdują się w pokoju. Robot jest na podłodze, skrzynka nadaje się do przeniesienia, wyłącznik jest wyłączony. Robot potrafi podejść do skrzynki lub wyłącznika, przenieść skrzynkę w inne miejsce, wejść na skrzynkę oraz załączyć wyłącznik, po uprzednim wejściu na skrzynkę (wyłącznik jest zbyt wysoko, by robot mógł go załączyć stojąc na podłodze).

W opisie zadania użyto predykatów, których sens wynika z nazwy predykatu.

$$\begin{aligned} S_p = \{ & (f_1) \text{ W(ROBOT, POKÓJ)}, \\ & (f_2) \text{ W(SKRZYNKA, POKÓJ)}, \\ & (f_3) \text{ W(WYŁĄCZNIK, POKÓJ)}, \\ & (f_4) \text{ NA PODŁODZE(ROBOT)}, \\ & (f_5) \text{ DO PODNIESIENIA(SKRZYNKA)}, \\ & (f_6) \text{ STAN(WYŁĄCZNIK, WYŁĄCZONY)}, \end{aligned}$$

(f_7) TYP(SKRZYŃKA, BOX),

(f_8) TYP(WYŁĄCZNIK, SWITCH) }

$S_k = \{(f_1) \text{ STAN(WYŁĄCZNIK, ZAŁĄCZONY)}\}$

$O = \{O_1, O_2, O_3, O_4\}$

O_1 - "Robot podchodzi do obiektu q_2 w pomieszczeniu q_1 ":

$Co = \{(co_1) \text{ NA PODŁODZE(ROBOT)},$

$(co_2) \text{ W(ROBOT, } q_1),$

$(co_3) \text{ W}(q_2, q_1)\}$

$Ca = \{(ca_1) \text{ PRZY(ROBOT, } q_3)\}$

$Ad = \{(ad_1) \text{ PRZY(ROBOT, } q_2)\}$

O_2 - "Robot przenosi obiekt q_1 do obiektu q_2 w pomieszczeniu q_3 ":

$Co = \{(co_1) \text{ TYP}(q_1, \text{BOX}),$

$(co_2) \text{ NA PODŁODZE(ROBOT)},$

$(co_3) \text{ PRZY(ROBOT, } q_1),$

$(co_4) \text{ W}(q_1, q_3),$

$(co_5) \text{ W}(q_2, q_3),$

$(co_6) \text{ DO PODNIESIENIA}(q_1)\}$

$Ca = \{(ca_1) \text{ PRZY(ROBOT, } q_4),$

$(ca_2) \text{ PRZY}(q_1, q_4)\}$

$Ad = \{(ad_1) \text{ PRZY}(q_1, q_2),$

$(ad_2) \text{ PRZY(ROBOT, } q_1)\}$

O_3 - "Robot wchodzi na obiekt q_1 ":

$Co = \{(co_1) \text{ PRZY(ROBOT, } q_1),$

$(co_2) \text{ NA PODŁODZE(ROBOT)},$

$(co_3) \text{ TYP}(q_1, \text{BOX})\}$

$Ca = \{(ca_1) \text{ NA PODŁODZE(ROBOT)}\}$

$Ad = \{(ad_1) \text{ NA(ROBOT, } q_1)\}$

O_4 - "Robot, stojąc na obiekcie q_1 , załącza wyłącznik q_2 ":

$$Co = \{(co_1) \text{ PRZY}(q_1, q_2), \\ (co_2) \text{ TYP}(q_2, \text{SWITCH}), \\ (co_3) \text{ NA}(\text{ROBOT}, q_1)\}$$

$$Ca = \{(ca_1) \text{ STAN}(q_2, \text{WYŁĄCZONY})\}$$

$$Ad = \{(ad_1) \text{ STAN}(q_2, \text{ZAŁĄCZONY})\}$$

W każdej z podanych reguł zbiór Re jest pusty ($Re = \emptyset$).

Z treści S_k wynika, że celem głównym w zadaniu jest znalezienie sekwencji reguł przekształcania, prowadzącej do załączenia wyłącznika.

4. OPIS PODSTAWOWYCH AKCJI, WYSTĘPUJĄCYCH W METODZIE

Opisywana w rozprawie metoda ma zapewnić znalezienie sekwencji przekształcającej $\mathcal{T}$ (p. 3.1). W tym celu wykorzystuje ona pewne akcje, których oddzielne omówienie ułatwi prezentację metody.

4.1. Tworzenie opisów nowych sytuacji

Podstawową akcją jest dokonywanie przekształcenia $S_i \xrightarrow{O_1} S_j$, za pomocą reguły przekształcania O_1 .

Definicja 4. Wektorem podstawienia Q_1^* dla reguły O_1 i sytuacji S_i (lub krótko: wektorem podstawienia Q_1^*) nazywamy uporządkowany zbiór

$$Q_1^* = (q_1^*, q_2^*, \dots, q_k^*),$$

zawierający stałe lub zmienne zwane podstawieniami, odpowiadające kolejnym elementom zbioru $Q_1 = (q_1, q_2, \dots, q_k)$ argumentów reguły O_1

$$(q_i^* \longleftrightarrow q_i), i=1, 2, \dots, k,$$

w sposób, zapewniający możliwość przyporządkowania każdej formule ze zbioru $Co \in O_1$ równoważnej formuły ze zbioru S_i . Przyporządkowanie to musi być wzajemnie jednoznaczne. Dopuszcza się jednak brak podstawień dla tych argumentów reguły O_1 , które występują tylko w zbiorze $Ca \in O_1$. Odpowiednie składowe wektora Q_1^* są wówczas równe tym argumentom.

Czynność wyznaczania wektora Q_1^* (mająca sens badania warunków użycia reguły O_1 do przekształcenia sytuacji S_i) musi uwzględniać ewentualne związki między argumentami $q_i \in Q_1$, zawarte w zbiorze $Re \in O_1$.

W ogólnym przypadku dla pewnej sytuacji S_i oraz pewnej reguły przekształcania O_1 może istnieć m wektorów podstawień:

$$Q_1^{*1}, Q_1^{*2}, \dots, Q_1^{*m}, m \geq 1.$$

Przekształcenie $O_1: S_i \xrightarrow{O_1} S_j$ za pomocą reguły O_1 może mieć miejsce, jeśli istnieje $m \geq 1$ wektorów podstawienia Q_1^* . Efektem użycia reguły O_1 do sytuacji S_i jest utworzenie m zbiorów formuł $S_j^1, S_j^2, \dots, S_j^m$, opisujących m nowych sytuacji.

Założmy istnienie jednego wektora podstawienia Q_1^* .

Oznaczmy:

- podzbiór argumentów, występujących w formule $co_i \in Co$, przez $Q_{1,i}, Q_{1,i} \in Q_1$, a tę formułę co_i przez $co_i(Q_{1,i})$;
- podzbiór argumentów, występujących we wszystkich formułach zbiorów Ca i Ad , odpowiednio przez $Q_{1,Ca}, Q_{1,Ca} \in Q_1$ i $Q_{1,Ad}, Q_{1,Ad} \in Q_1$, a zbiory Ca i Ad przez $Ca(Q_{1,Ca})$ i $Ad(Q_{1,Ad})$;
- formułę $co_i(Q_{1,i})$, po dokonaniu podstawień $Q_{1,i} := Q_{1,i}^*$ elementów wektora Q_1^* pod odpowiednie argumenty wektora Q_1 , przez $co_i(Q_{1,i}^*)$;
- zbiory $Ca(Q_{1,Ca})$ i $Ad(Q_{1,Ad})$, po dokonaniu podstawień jw., przez $Ca(Q_{1,Ca}^*)$ i $Ad(Q_{1,Ad}^*)$.

Wówczas użycie reguły O_1 do przekształcenia sytuacji S_i można zapisać jako implikację:

$$\bigvee_{Q_1^*} : \left\{ \bigwedge_{co_i(Q_{1,i}) \in Co} \bigvee_{f_{i_k} \in S_i, k \in \{1,2,\dots,n\}} : f_{i_k} \equiv co_i(Q_{1,i}^*) \right\} \Rightarrow \\ \Rightarrow \bigvee_{S_j} : \left\{ O_1: S_i \xrightarrow{O_1} S_j, S_j = S_i \setminus Ca(Q_{1,Ca}^*) \cup Ad(Q_{1,Ad}^*) \right\}.$$

Z powyższego zapisu wynika, że opis nowej sytuacji S_j powstaje na podstawie zbioru S_i , poprzez wyłączenie ze zbioru S_i formuł równoważnych z formułami zbioru $Ca \in O_1$ (po uprzednim dokonaniu w formułach zbioru Ca podstawień elementów Q_1^* pod odpowiednie argumenty Q_1) oraz poprzez dołączenie do zbioru S_i formuł zbioru $Ad \in O_1$ (po dokonaniu w nich podstawień elementów Q_1^* pod odpowiednie argumenty Q_1).

Dla m wektorów podstawienia $Q_1^{*1}, Q_1^{*2}, \dots, Q_1^{*m}$ podana wyżej implikacja przyjmuje postać:

$$\bigwedge_{r \in \{1, 2, \dots, m\}} \left[\bigvee_{Q_1^{*r}} \left\{ \bigwedge_{\text{co}_i(Q_{1,i}) \in \text{Co}} \bigvee_{f_{i_k} \in S_i, k \in \{1, 2, \dots, n\}} : f_{i_k} \equiv \text{co}_i(Q_{1,i}^{*r}) \right\} \Rightarrow \right. \\ \left. \Rightarrow \bigvee_{S_j} \left\{ 0_1 : S_i \xrightarrow{0_1} S_j^r, S_j^r = S_i \setminus \text{Ca}(Q_{1,\text{Ca}}^{*r}) \cup \text{Ad}(Q_{1,\text{Ad}}^{*r}) \right\} \right].$$

Przekształcenie $S_i \xrightarrow{0_1} S_j$ można przedstawić za pomocą grafu skierowanego o strukturze drzewa (rys. 4.1).

W ogólnym przypadku sytuację S_i można przekształcić za pomocą kilku reguł przekształcania ze zbioru O . Każda z nowo utworzonych sytuacji może być z kolei podstawą do utworzenia kolejnych, nowych sytuacji. Taki zbiór przekształceń, dokonywanych za pomocą reguł ze zbioru O na sytuacji początkowej S_p i sytuacjach utworzonych z S_p , można przedstawiać za pomocą drzewa z korzeniem S_p (rys. 4.2). Drzewo to będziemy nazywać drzewem rozwiązania (DR).

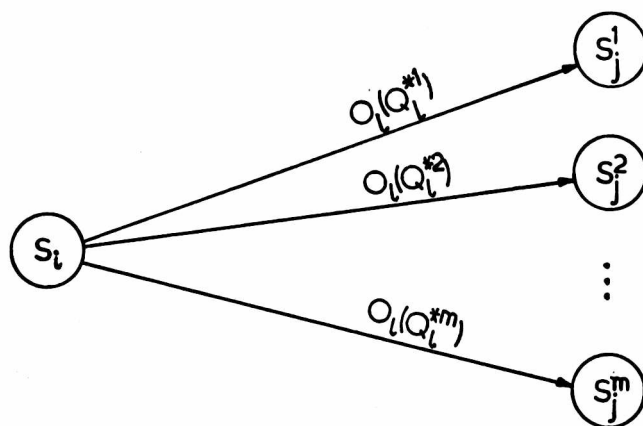
DR rozwija się zarówno "wszerz", jak i "w głąb". O rozwoju "wszerz" decyduje ilość możliwych wektorów podstawień, o rozwoju "w głąb" decyduje ilość przekształceń dokonanych kolejno.

Na rys. 4.3 przedstawiono przekształcenie sytuacji S_p z przykładu w p. 3.3, za pomocą reguły przekształcania 0_1 .

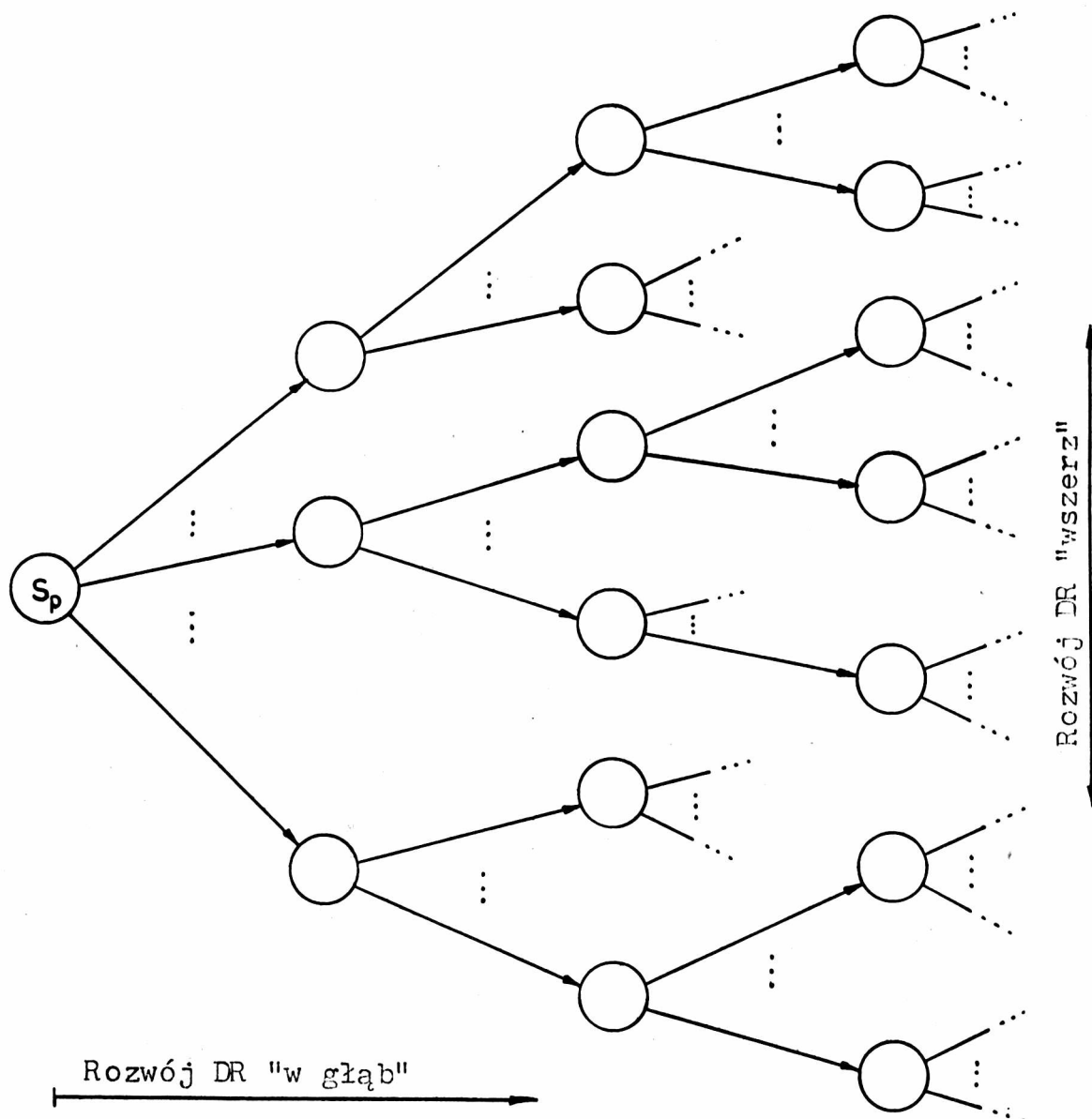
Opisane tu dokonywanie przekształceń $S_i \xrightarrow{0_1} S_j$ za pomocą reguły 0_1 będziemy nazywać przekształcaniem "wprost" i będziemy oznaczać przez PW.

4.2. Rozkład celu na podcele

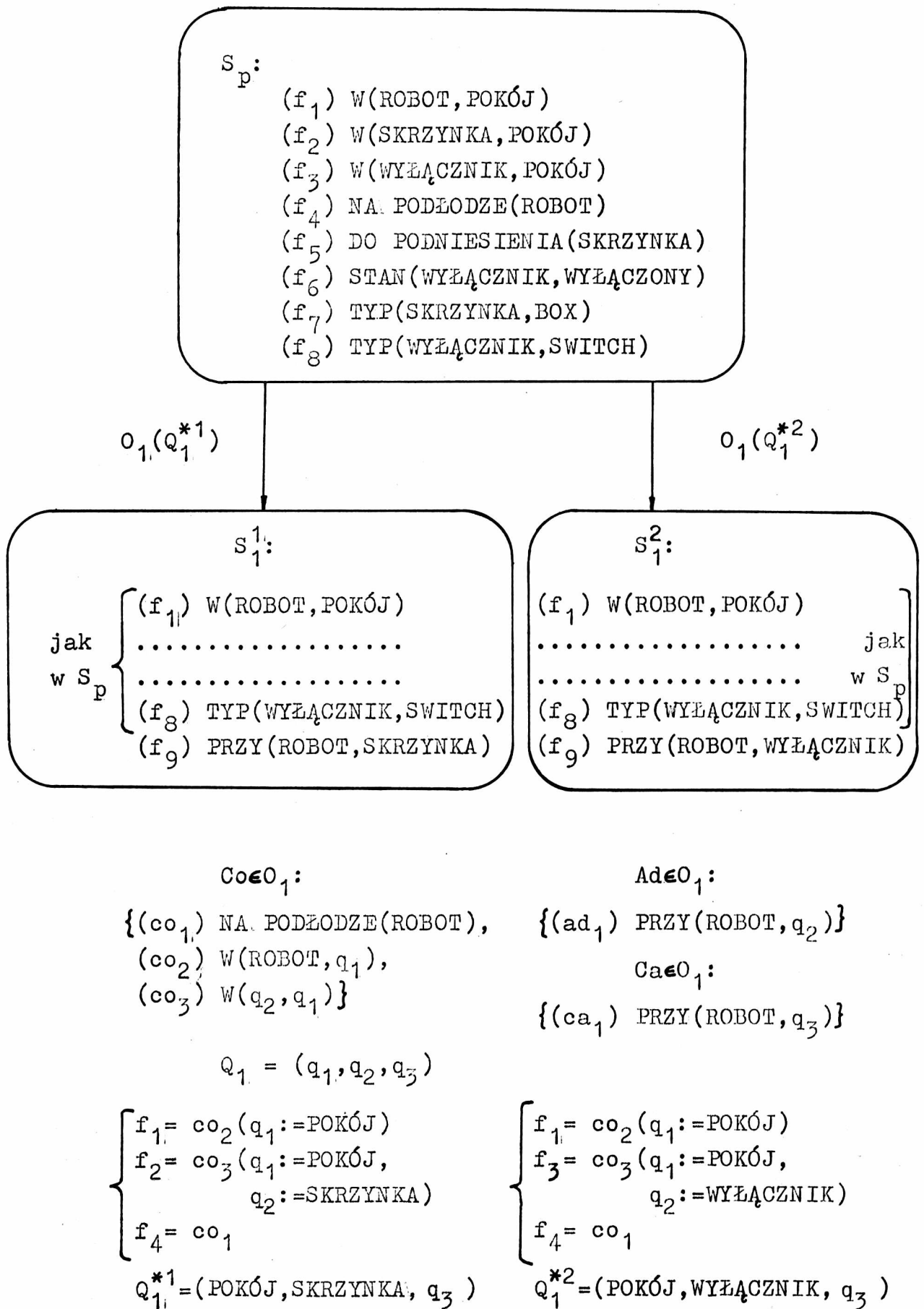
Reguły przekształcania, których użycie do dokonywania przekształceń PW opisano w p. 4.1, mogą być również zastosowane do dokonywania innego rodzaju przekształceń, tzw. przekształceń wstecznych, które będziemy oznaczać przez PO. Idea tych przekształceń wynika ze spostrzeżenia, że jeśli regułę



Rys. 4.1. Przekształcenie $S_i \xrightarrow{O_l} S_j$, zrealizowane przy użyciu reguły przekształcania O_l , z wektorami podstawienia $Q_l^{*1}, Q_l^{*2}, \dots, Q_l^{*m}$.



Rys. 4.2. Drzewo rozwiązania (DR) - schemat rozwoju.



Rys. 4.3. Przykład przekształcenia PW. Przekształcenie sytuacji S_p za pomocą reguły przekształcania O_1 (p.3.3)

O_1 można w uproszczeniu traktować jako transformację pewnych stanów $b_1^i, b_2^i, \dots, b_p^i$ zbioru B w inne stany $b_1^j, b_2^j, \dots, b_s^j$, to dla danego stanu $b_k^j, k \in \{1, 2, \dots, s\}$ można określić za pomocą O_1 zbiór stanów $b_1^i, b_2^i, \dots, b_p^i$, których przekształcenie PW za pomocą O_1 tworzy stany, wśród których znajduje się stan b_k^j .

Ponieważ elementy przestrzeni $\mathcal{P}$ opisywane są za pomocą formuł, przekształcenia wsteczne PO będziemy przedstawiać jako przekształcenia formuł.

Definicja 5. Wektorem podstawienia V_1^* dla reguły O_1 i formuły f (lub krótko: wektorem podstawienia V_1^*) nazywamy uporządkowany zbiór

$$V_1^* = (v_1^*, v_2^*, \dots, v_k^*),$$

zawierający stałe lub zmienne zwane podstawieniami, które podstawione w miejsce kolejnych argumentów ze zbioru $Q_1 = (q_1, q_2, \dots, q_k)$ reguły O_1

$$(q_i := v_i^*), i=1, 2, \dots, k$$

doprowadzają do równoważności jednej z formuł ze zbioru $Ad \in O_1$ z formułą f .

Argumenty reguły O_1 , występujące tylko w zbiorze $Ca \in O_1$, nie biorą udziału w przekształceniu i tym samym odpowiednie składowe V_1^* są równe tym argumentom. Pod argumenty reguły O_1 , występujące tylko w zbiorze $Co \in O_1$, podstawia się nowe zmienne $x_i, i=1, 2, \dots$ (będą one później pełnić rolę argumentów drzewa podcelów DP).

Czynność wyznaczania wektora V_1^* musi uwzględniać ewentualne związki między argumentami $q_i \in Q_1$, zawarte w zbiorze $Re \in O_1$.

W ogólnym przypadku dla pewnej formuły f oraz pewnej reguły przekształcania O_1 może istnieć m wektorów podstawień:

$$V_1^{*1}, V_1^{*2}, \dots, V_1^{*m}, m \geq 1.$$

Założmy istnienie jednego wektora podstawienia V_1^*

Oznaczmy:

- podzbiór argumentów, występujących w formule $ad_k \in Ad$, przez $Q_{1,k}$, $Q_{1,k} \in Q_1$;
- formułę ad_k przez $ad_k(Q_{1,k})$;
- formułę $ad_k(Q_{1,k})$, po dokonaniu podstawień elementów wektora V_1^* pod odpowiednie argumenty wektora Q_1 , przez $ad_k(V_{1,k}^*)$;
- formułę $co_i \in Co$, po dokonaniu podstawień elementów wektora V_1^* pod odpowiednie argumenty wektora Q_1 , przez $co_i(V_{1,i}^*)$.

Wprowadźmy ponadto pojęcie celu. Celem p będziemy nazywać element przestrzeni $\mathcal{P}$, opisany formułą f .

Użycie reguły O_1 do przekształcenia wstecznego formuły f przy ustalonym wektorze V_1^* można zapisać jako implikację:

$$V_1^* : \left\{ \bigvee_{k \in \{1,2,\dots,s\}} : f \equiv ad_k(V_{1,k}^*) \right\} \Rightarrow \{f_1, f_2, \dots, f_p\} : \\ \left[f \rightarrow \{f_1, f_2, \dots, f_p\}, \bigwedge_{i \in \{1,2,\dots,p\}} f_i := co_i(V_{1,i}^*) \right].$$

Zbiór formuł $\{f_1, f_2, \dots, f_p\}$, otrzymanych ze zbioru formuł Co reguły O_1 przez podstawienie w miejsce ich argumentów odpowiednich podstawień - składowych wektora V_1^* , nazywamy wynikiem przekształcenia wstecznego formuły f za pomocą reguły O_1 i ustalonego wektora podstawienia V_1^* . Formuły te, zawierające w ogólnym przypadku zmienne x_i , $i=1,2,\dots$, opisują rodziny elementów przestrzeni stanów $\mathcal{P}$. Przy ustalonych wartościach zmiennych x_i , $i=1,2,\dots$ formuły opisują konkretne elementy przestrzeni $\mathcal{P}$. Elementy $p_1, p_2, \dots, p_p$ przestrzeni $\mathcal{P}$, opisane przez formuły $f_1, f_2, \dots, f_p$, będziemy nazywać podcelami. Dla podkreślenia uzależnienia podcelów $p_1, p_2, \dots, p_p$ od celu p , będziemy nazywać p celem nadrzędnym względem podcelów $p_1, p_2, \dots, p_p$.

Przekształcenie wsteczne PO za pomocą reguły O_1 przy usta-

lonym wektorze V_1^* , będziemy również oznaczać przez $O_1^{-1}(V_1^*)$ i będziemy nazywać rozkładem (lub rozbiciem) celu p na podcele $p_1, p_2, \dots, p_p$.

Wystąpienie podcelów $p_1, p_2, \dots, p_p$ w pewnej sytuacji S umożliwia otrzymanie nowej sytuacji, zawierającej cel p , w wyniku przekształcenia PW sytuacji S za pomocą reguły O_1 (będzie o tym mowa w dalszej części rozprawy). Sytuacji, zawierających podcele $p_1, p_2, \dots, p_p$ może być w przestrzeni $\mathcal{P}$ wiele.

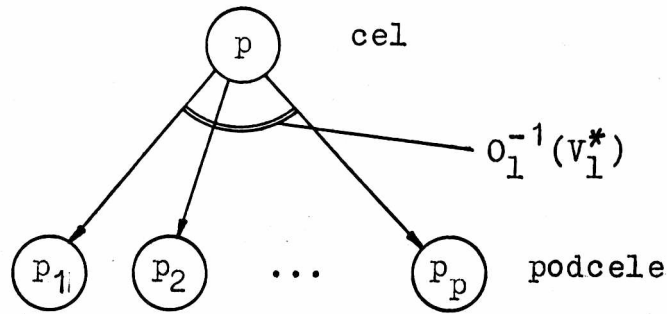
Przekształcenie PO można przedstawić za pomocą grafu skierowanego o strukturze drzewa (rys. 4.4). W grafie tym wierzchołek początkowy p reprezentuje cel p , opisany formułą f , natomiast wierzchołki końcowe $p_1, p_2, \dots, p_p$ reprezentują podcele, opisane formułami $f_1, f_2, \dots, f_p$. Wszystkie te podcele związane są równocześnie z celem p i regułą O_1 , która posłużyła do ich utworzenia. Dla podkreślenia logicznego związku między podcelami $p_1, p_2, \dots, p_p$, będziemy je nazywać podcelami koniunkcyjnymi. W grafie, obrazującym przekształcenie PO, będziemy łączyć łukiem gałęzie, łączące cel p z podcelami koniunkcyjnymi.

Na rys. 4.5 przedstawiono rozkład celu p_1 , opisanego formułą $f_1 \in S_k$ z przykładu w p. 3.3, na podcele, za pomocą reguły O_4 .

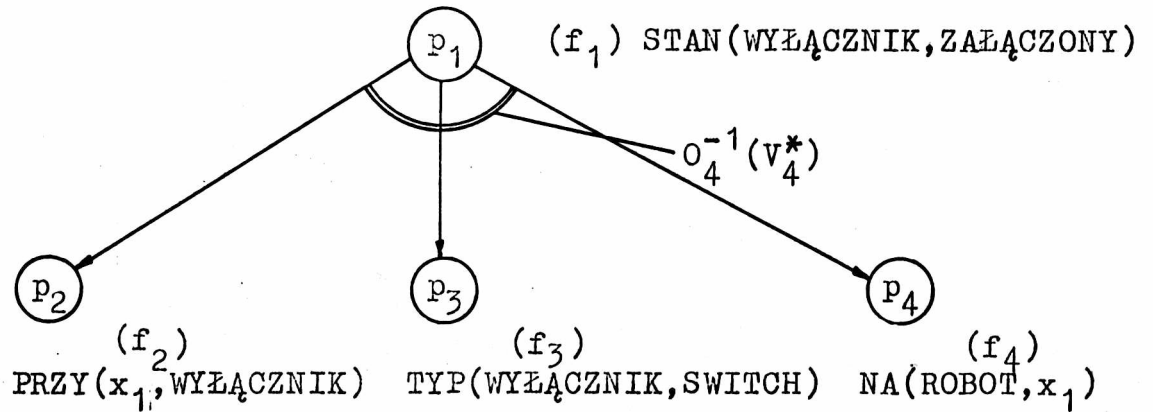
Dla m wektorów podstawienia $V_1^{*1}, V_1^{*2}, \dots, V_1^{*m}$ implikacja, określająca przekształcenie PO, przyjmuje postać:

$$\bigwedge_{r \in \{1, 2, \dots, m\}} : \left\{ V_1^{*r} : \left[\bigvee_{k \in \{1, 2, \dots, s\}} : f \equiv \text{ad}_k(V_1^{*r}) \right] \Rightarrow \{f_1^r, f_2^r, \dots, f_p^r\} : \right. \\ \left. \left[f \rightarrow \{f_1^r, f_2^r, \dots, f_p^r\}, \bigwedge_{i \in \{1, 2, \dots, p\}} f_i^r := \text{co}_i(V_1^{*r}) \right] \right\}.$$

W tym przypadku powstaje m zbiorów podcelów koniunkcyj-



Rys. 4.4. Rozkład celu p na podcele $p_1, p_2, \dots, p_p$, zrealizowany przy użyciu reguły przekształcania O_1 i wektora podstawienia V_1^* .



$Ad \in O_4:$

$\{(ad_1) \text{ STAN}(q_2, \text{ZAŁĄCZONY})\}$

$Co \in O_4:$

$\{(co_1) \text{ PRZY}(q_1, q_2),$
 $(co_2) \text{ TYP}(q_2, \text{SWITCH}),$
 $(co_3) \text{ NA}(\text{ROBOT}, q_1)\}$

$Q_4 = (q_1, q_2)$

$f_1 \equiv ad_1(q_2 := \text{WYŁĄCZNIK})$

$V_4^* = (x_1, \text{WYŁĄCZNIK})$

Rys. 4.5. Przykład przekształcenia PO. Rozkład celu $p_1 \equiv S_k$ na podcele p_2, p_3, p_4 , za pomocą reguły przekształcania O_4 (p. 3.3).

nych celu p. Poszczególne zbiory podcelów stanowią odrębne zbiory warunków, które muszą zaistnieć, by mógł być osiągnięty cel p.

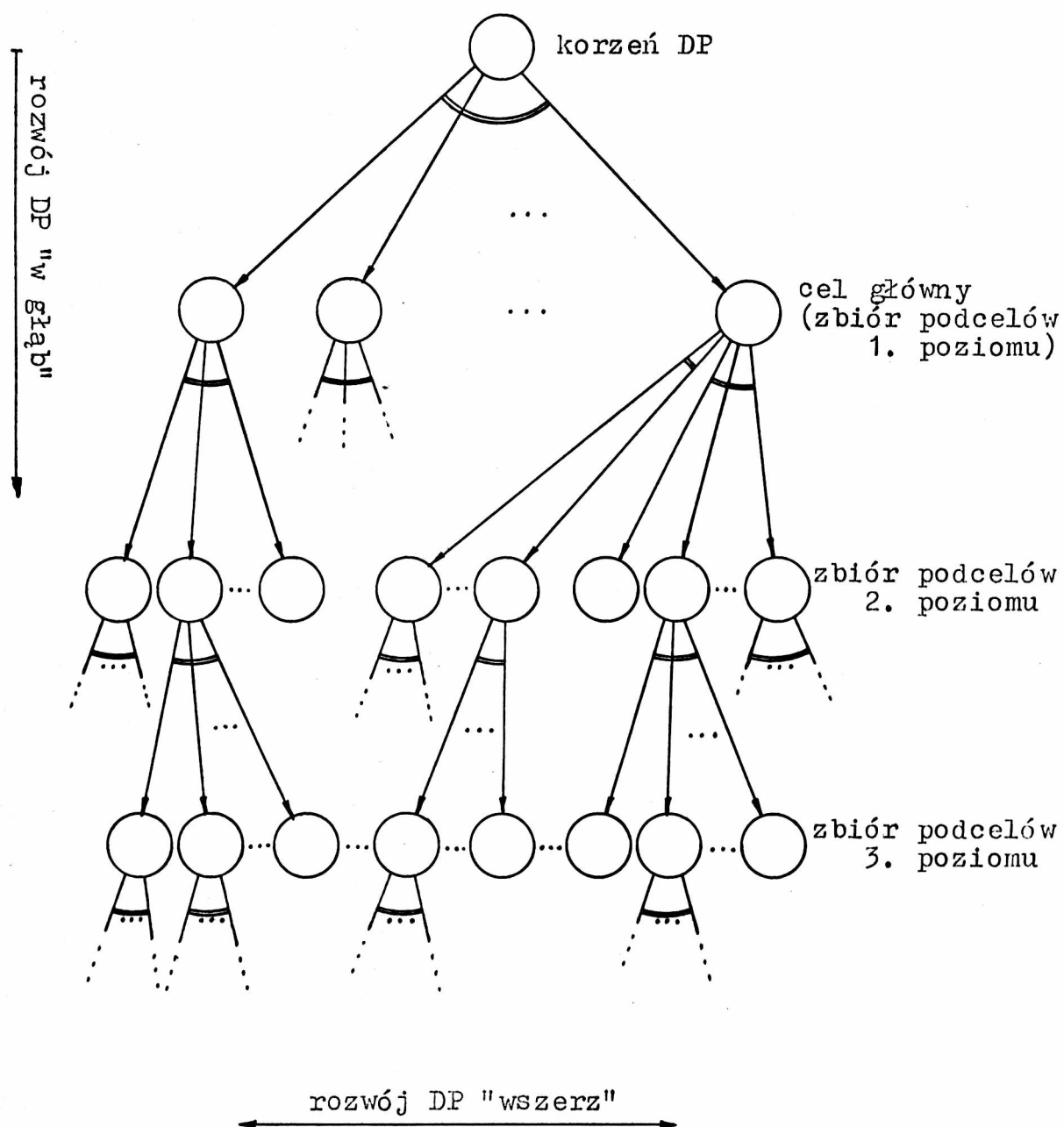
W ogólnym przypadku cel p może być rozłożony na podcele za pomocą kilku reguł przekształcania ze zbioru O. Z kolei każdy utworzony podcel może być rozbijany, jako cel, na podcele, itd.

Określony w zadaniu (p. 3.1) podzbiór S_k sytuacji docelowej S_d , opisany zbiorem formuł S_k , będzie w dalszych opisach nazywany celem głównym. Skupia on w koniunkcję cele, które mają być osiągnięte za pomocą poszukiwanej sekwencji przekształcającej $\mathcal{T}$. Każdy z tych celów można w ogólnym przypadku wieloetapowo rozłożyć na podcele składowe, przy użyciu różnych reguł O_1 i różnych wektorów V_1^* . Taki zbiór przekształceń PO można przedstawić za pomocą drzewa z korzeniem, skupiającym w koniunkcję cele, opisane zbiorem S_k formuł (rys. 4.6). Drzewo to będziemy nazywać drzewem podcelów (DP). Ma ono charakter drzewa (grafu) typu "i-lub" [11], [26], [49].

W DP istnieje wielopoziomowa hierarchia podcelów: cel główny występuje na najwyższym, 1. poziomie (tworzy on zbiór koniunkcyjny tzw. podcelów 1. poziomu), podcele każdego z następnych poziomów (2,3,...) są podporządkowane odpowiednim podcelom poprzedniego, wyższego poziomu.

W opisach podcelów w DP, na wszystkich poziomach poza poziomem 1., mogą pojawiać się zmienne x_i , $i=1,2,\dots$. Wprowadzane są one bądź przez kolejne przekształcenia PO (wówczas zmienne te otrzymują kolejne numery), bądź przenoszone są z danego poziomu DP na poziom niższy, jako składowe wektora V_1^* , podczas rozkładu celu, zawierającego w opisie zmienne x_i , na podcele.

Drzewo podcelów DP, rozbudowane do pewnej postaci, zawiera



Rys. 4.6. Drzewo podcelów DP - schemat rozwoju

w opisach podcelów zmienne x_i , $i=1,2,\dots,n$, tworzące uporządkowany zbiór

$$X = (x_1, x_2, \dots, x_n),$$

który będziemy nazywać zbiorem argumentów DP.

Dla każdej zmiennej x_i , $i=1,2,\dots,n$, wyrażającej obiekt lub stan, istnieje zbiór dopuszczalnych wartości. Uporządkowany zbiór

$$X^* = (x_1^*, x_2^*, \dots, x_n^*),$$

gdzie x_i^* jest dopuszczalną wartością zmiennej x_i , będziemy dalej nazywać wektorem podstawienia X^* . Wektor X^* , podstawiony w miejsce zbioru X , sprawia, że DP zawiera zbiór konkretnych elementów przestrzeni stanów $\mathcal{P}$ i wyraża wzajemne związki między nimi. Będziemy ponadto rozpatrywać indeksowane zbiory X_j^* (indeks j otrzyma interpretację w dalszej części opisu), w których pewne elementy x_i^* są ustalonymi wartościami zmiennych x_i , natomiast pozostałe elementy są nieokreślone. Każdy z takich zbiorów X_j^* przedstawia rodzinę wektorów podstawień.

Założmy istnienie DP, którego fragmentem są: cel p oraz zbiór podcelów koniunkcyjnych $p_1, p_2, \dots, p_p$, utworzonych przez rozkład celu p , za pomocą reguły O_1 i wektora V_1^* . Niech ponadto istnieją: sytuacja S_i , opisana zbiorem S_i formuł oraz zbiór X_j^* taki, że po podstawieniu go w miejsce zbioru X możliwe jest wzajemnie jednoznaczne przyporządkowanie każdemu podcelowi $p_1, p_2, \dots, p_p$ równoważnej mu formuły z opisu sytuacji S_i . Wówczas można sytuację S_i przekształcić wprost (PW), za pomocą reguły O_1 , w nową sytuację S_j , zawierającą cel p , nadrzędny względem podcelów $p_1, p_2, \dots, p_p$. Wektor Q_1^* podstawienia pod argumenty Q_1 reguły O_1 (p. 4.1), umożliwiający to przekształcenie PW, tworzy się poprzez:

- podstawienie elementów wektora V_1^* pod odpowiadające im argumenty ze zbioru Q_1 ;
- podstawienie pod zmienne $x_i \in X$ elementów wektora X_j^* ;
- wyznaczenie podstawień pod te argumenty ze zbioru Q , które nie otrzymały podstawień z V_1^* i X_j^* (zgodnie z zasadą przekształcenia PW).

Indeks j zbioru X_j^* zwiążemy z indeksem nowoutworzonej sytuacji S_j , dla podkreślenia faktu uzależnienia sytuacji S_j od postaci zbioru X_j^* .

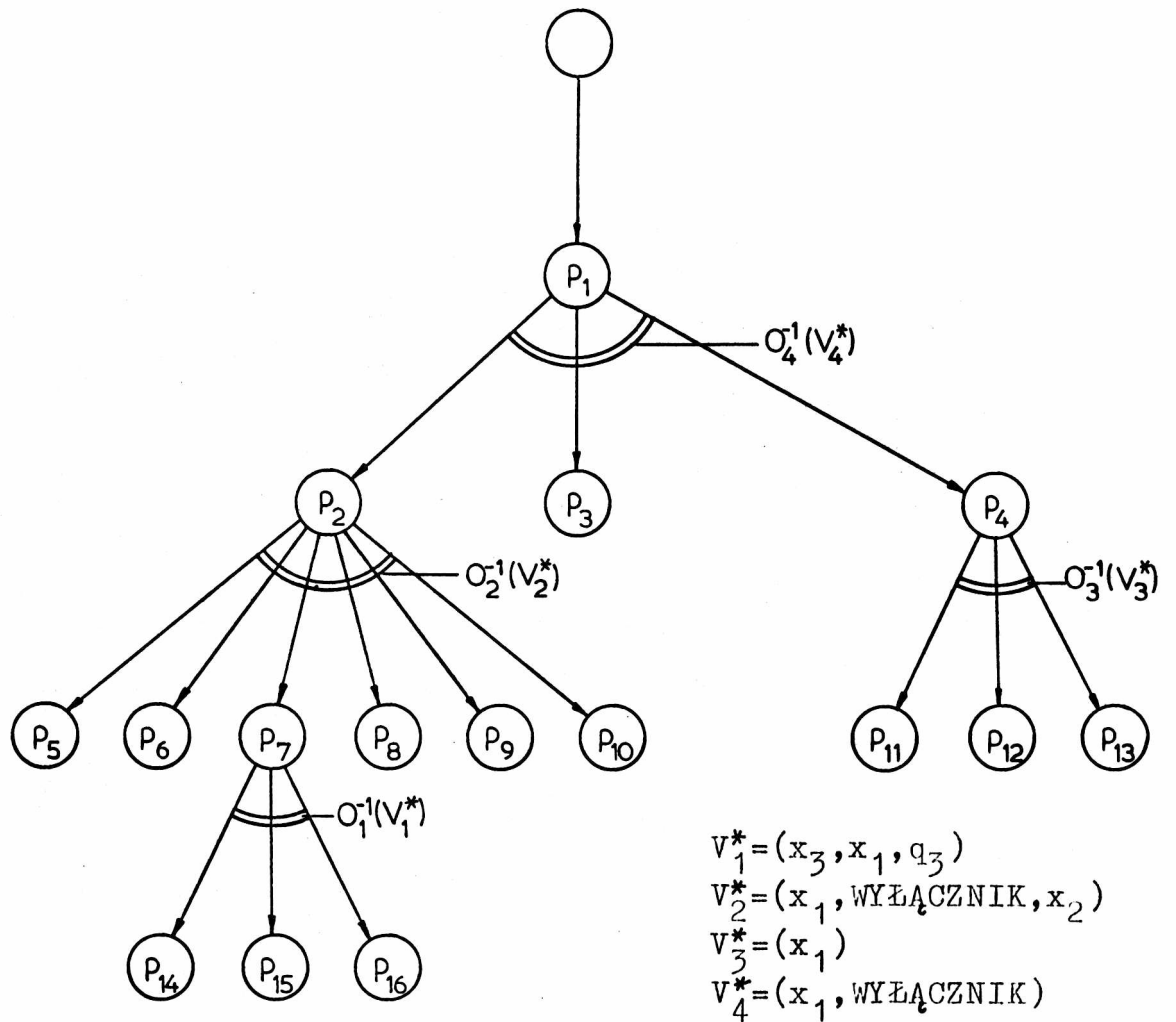
Z powyższego opisu wynika, że fragment DP, złożony ze zbioru podcelów $p_1, p_2, \dots, p_p$ (zawartych w sytuacji S_i), celu p (odpowiadającego nowej sytuacji S_j) oraz reguła O_1 , "ukierunkowuje" przekształcenie $S_i \xrightarrow{O_1} S_j$ na tworzenie sytuacji, zawierających cel p (czyli na osiągnięcie celu p). Można więc ten fragment DP uważać za uproszczony schemat osiągnięcia celu p . Całe DP można natomiast uważać za schematyczne przedstawienie wielu wariantów przekształceń PW, umożliwiających osiągnięcie celu głównego.

Drzewo podcelów DP może rozwijać się zarówno "wszerz", jak i "w głąb". O rozwoju "wszerz" decyduje ilość możliwych rozbić celu na podcele, o rozwoju "w głąb" decyduje ilość przekształceń PO dokonanych kolejno na danej ścieżce w DP.

Na rys. 4.7 przedstawiono fragment DP, utworzonego przez rozkład celu, opisanego zbiorem S_k z p. 3.3, na podcele, za pomocą reguł przekształcania ze zbioru O .

4.3. Analiza drzewa podcelów i budowa drzewa rozwiązania

Działanie metody poszukiwania sekwencji przekształcającej $\mathcal{T}$ polega na systematycznej, ukierunkowywanej pewną strategią, budowie drzewa rozwiązania DR, aż do uzyskania wierzchołka S_d , odpowiadającego sytuacji docelowej S_d , której podzbiorem



Wektor $X = (x_1, x_2, x_3)$

- | | |
|---|--|
| (p ₁) STAN(WYŁĄCZNIK, ZAŁĄCZONY) | (p ₉) W(WYŁĄCZNIK, x ₂) |
| (p ₂) PRZY(x ₁ , WYŁĄCZNIK) | (p ₁₀) DO PODNIESIENIA(x ₁) |
| (p ₃) TYP(WYŁĄCZNIK, SWITCH) | (p ₁₁) PRZY(ROBOT, x ₁) |
| (p ₄) NA(ROBOT, x ₁) | (p ₁₂) NA PODŁODZE(ROBOT) |
| (p ₅) TYP(x ₁ , BOX) | (p ₁₃) TYP(x ₁ , BOX) |
| (p ₆) NA PODŁODZE(ROBOT) | (p ₁₄) NA PODŁODZE(ROBOT) |
| (p ₇) PRZY(ROBOT, x ₁) | (p ₁₅) W(ROBOT, x ₃) |
| (p ₈) W(x ₁ , x ₂) | (p ₁₆) W(x ₁ , x ₃) |

Cel główny: p₁

Rys. 4.7. Fragment drzewa podcelów, utworzonego przez rozbięcie celu S_k z przykładowego zadania w p. 3.3 na podcele, za pomocą reguł przekształcania ze zbioru 0

jest zadany opis S_k , określony w zadaniu (p. 3.1). Wówczas ścieżka w DR, od wierzchołka początkowego S_p (korzenia DR) do wierzchołka S_d , odpowiada poszukiwanej sekwencji przekształcającej $\mathcal{T}$.

Budowa drzewa rozwiązania bez ukierunkowywania jego rozwoju, czyli wykonywanie wszystkich możliwych przekształceń PW sytuacji w każdym wierzchołku drzewa, oznaczałoby poszukiwanie sytuacji docelowej metodą pełnego przeglądu wszystkich wariantów przekształceń, w tym również bardzo wielu wariantów nie zbliżających do celu. Takie "szukanie na oślep" nie ma praktycznie sensu w większości problemów, głównie ze względu na ogromną czasochłonność oraz bardzo dużą ilość informacji, którą należy pamiętać podczas takiego działania.

Ogólną strategią sterowania budową drzewa rozwiązania DR jest uzależnienie kierunku jego rozwoju od postaci drzewa podcelów DP. Podstawową akcją jest tu tzw. analiza drzewa podcelów DP, połączona z rozbudową DR.

Dla zobrazowania tego rodzaju akcji przyjmijmy, że:

- istnieje DP, rozbudowane do pewnej postaci (co najmniej jedno rozbitcie celu p na podcele $p_1, p_2, \dots, p_p$, za pomocą reguły O_1),
- istnieje DR, rozbudowane do pewnej postaci (może być tylko wierzchołek początkowy S_p drzewa DR, reprezentujący sytuację początkową S_p).

Zapoczątkowanie procesu analizy DP i budowy DR wymaga wcześniejszego wybrania podcelu p_k (w DP), od którego ma się rozpocząć analiza DP, oraz sytuacji S_i (w DR), względem której analiza ta będzie przeprowadzana.

Założmy, że mamy przeprowadzić analizę DP od pewnego podcelu p_k , względem sytuacji S_i . Podcel p_k jest elementem pewnego zbioru P podcelów koniunkcyjnych w DP

$$P = \{p_1, p_2, \dots, p_k, \dots, p_p\}.$$

Istnieje również określony cel nadrzędny p względem zbioru P podcelów koniunkcyjnych oraz reguła przekształcania O_1 , za pomocą której utworzono wcześniej zbiór P (p. 4.2).

Analiza DP i budowa DR jest procesem rekurencyjnym, wykorzystującym dwa rodzaje tzw. kroków analizy: kroki pionowe oraz kroki poziome. Krok pionowy możliwy jest wówczas, gdy w opisie sytuacji S_i znajdują się formuły równoważne (z uwzględnieniem podstawień pod argumenty x_i drzewa podcelów DP) podcelom $p_1, p_2, \dots, p_p$. Wówczas jest możliwe przekształcenie sytuacji S_i w nową sytuację S_j , zawierającą cel nadrzędny p (p. 4.2). Powstaje nowa sytuacja S_j a w przekształceniu $S_i \xrightarrow{O_1} S_j$ bierze udział pewien zbiór X_j^* podstawień pod argumenty DP w podcelach $p_1, p_2, \dots, p_p$ oraz podstawień dokonanych wcześniej, przy przekształceniach prowadzących do S_i . Zbiór ten będzie odtąd związany z sytuacją S_j , jako zbiór podstawień, warunkujący utworzenie danej sytuacji S_j oraz ewentualnych następnych sytuacji, uzyskiwanych w wyniku przekształcania S_j .

Sytuacji S_j odpowiada w DP cel p , leżący bliżej celu głównego niż wierzchołek $p_k \in P$. Nastąpiło zatem przemieszczenie na wyższy poziom w DP (stąd nazwa kroku: pionowy). Dalsze przemieszczanie na kolejny, wyższy poziom DP jest możliwe, gdy w opisie sytuacji S_j znajdują się formuły równoważne (z uwzględnieniem zgodnych z X_j^* podstawień pod argumenty drzewa podcelów DP) podcelom koniunkcyjnym, do których należy cel p , osiągnięty w poprzednim kroku.

Pomyślnym przebiegiem analizy jest sekwencja kroków pionowych, natomiast sukces analizy polega na osiągnięciu celu głównego w DP. Niepomyślny przebieg analizy ma miejsce wówczas, gdy krok pionowy nie jest możliwy, tzn. gdy opis sytuacji zawiera formuły równoważne (z uwzględnieniem podstawień)

jedynie niektórym podcelom ze zbioru P (oznaczymy podzbiór tych podcelów przez P^+ , $P^+ \subset P$), natomiast dla pozostałych podcelów (podzbiór P^- , $P^- = P \setminus P^+$) takie przyporządkowanie, zapewniające równowagę, nie istnieje. Wówczas należy podjąć próbę wykonania serii kroków poziomych, zmierzających do utworzenia pewnej sytuacji, w której wystąpią wszystkie podcele ze zbioru P (co umożliwi wykonanie kroku pionowego, zbliżającego do celu).

Do wykonania serii kroków poziomych przystępuje się tylko wówczas, gdy każdy podcel z P^- posiada w DP swoje rozbitcie na podcele. Osiąganie każdego z kolejnych podcelów $p_j \in P^-$ polega na wykonaniu przekształcenia PW, zwanego w tym przypadku krokiem poziomym, określonego fragmentem DP, reprezentującym rozkład p_j na podcele (o ile przekształcenie to jest wykonalne dla danej sytuacji S_i). Kolejno powstającym sytuacjom: S_j , S_{j+1} , S_{j+2} , ... nie odpowiada w tym wypadku przemieszczanie na kolejne, wyższe poziomy DP (jak przy wykonywaniu kroku pionowego), lecz osiąganie kolejnych podcelów, leżących w DP na tym samym poziomie. Każda z tych sytuacji powinna tylko zmniejszać ilość podcelów z P^- , aż do całkowitego zlikwidowania P^- w pewnej sytuacji S_m . Wówczas kończy się seria kroków poziomych i można wykonać wcześniej opisany krok pionowy dla sytuacji S_m .

Zarówno podczas wykonywania kroku pionowego, jak i podczas wykonywania kroku poziomego może okazać się, że dla sytuacji S_i oraz danego zbioru podcelów koniunkcyjnych P można znaleźć $m > 1$ zbiorów podstawień: x_j^{*1} , x_j^{*2} , ..., x_j^{*m} . Wówczas, zgodnie z zasadą przekształcenia PW (p. 4.1), powstaje m nowych sytuacji: S_j^1 , S_j^2 , ..., S_j^m , którym w DP odpowiada ten sam cel nadrzędny względem rozpatrywanych podcelów z P . W DR jest wówczas m gałęzi, odchodzących od wierzchołka S_i , z których każ-

da może prowadzić dalej do celu i w związku z tym powinna być analizowana i ewentualnie rozwijana dalej, w opisany wyżej sposób.

Niepowodzeniem analizy jest napotkanie zbioru podcelów koniunkcyjnych, którym nie można przyporządkować równoważnych formuł w opisie danej sytuacji oraz nie jest możliwe kolejne osiągnięcie tych podcelów w wyniku serii kroków poziomych.

Dla zilustrowania powyższych opisów rozpatrzmy przykład analizy DP z rys. 4.7, dla przykładowego zadania z p. 3.3.

PRZYKŁAD. Weźmy sytuację S_p oraz podcel p_{15} , należący do zbioru $P = \{p_{14}, p_{15}, p_{16}\}$ podcelów koniunkcyjnych, utworzonych na podstawie p_7 za pomocą reguły przekształcania O_1 .

$$\begin{aligned} S_p = \{ & (f_1) \text{ W(ROBOT, POKÓJ)}, \\ & (f_2) \text{ W(SKRYNKA, POKÓJ)}, \\ & (f_3) \text{ W(WYŁĄCZNIK, POKÓJ)}, \\ & (f_4) \text{ NA PODŁODZE(ROBOT)}, \\ & (f_5) \text{ DO PODNIESIENIA(SKRYNKA)}, \\ & (f_6) \text{ STAN(WYŁĄCZNIK, WYŁĄCZONY)}, \\ & (f_7) \text{ TYP(SKRYNKA, BOX)}, \\ & (f_8) \text{ TYP(WYŁĄCZNIK, SWITCH)} \} \end{aligned}$$

$$\begin{aligned} P = \{ & (p_{14}) \text{ NA PODŁODZE(ROBOT)}, \\ & (p_{15}) \text{ W(ROBOT, } x_3), \\ & (p_{16}) \text{ W}(x_1, x_3) \} \end{aligned}$$

Przyporządkowania:

$$\begin{aligned} (p_{14}) & \longleftrightarrow (f_4), \\ (p_{15}) & \longleftrightarrow (f_1), \\ (p_{16}) & \longleftrightarrow (f_2) \end{aligned}$$

pozwalają wydzielić zbiór podstawień x_1^*

$$x_1^* = \{x_1 := \text{SKRYNKA}, x_3 := \text{POKÓJ}\}$$

i przekształcić S_p w S_1 , za pomocą reguły przekształcania O_1 (jest to pierwszy krok pionowy).

$$S_1 = \{ (f_1) W(ROBOT, POKÓJ), \\ (f_2) W(SKRYNKA, POKÓJ), \\ (f_3) W(WYŁĄCZNIK, POKÓJ), \\ (f_4) NA PODŁODZE(ROBOT), \\ (f_5) DO PODNIESIENIA(SKRYNKA), \\ (f_6) STAN(WYŁĄCZNIK, WYŁĄCZONY), \\ (f_7) TYP(SKRYNKA, BOX), \\ (f_8) TYP(WYŁĄCZNIK, SWITCH), \\ (f_9) PRZY(ROBOT, SKRYNKA) \}$$

Utworzona sytuacja S_1 zawiera cel p_7 (po podstawieniu w p_7 : $x_1 := SKRYNKA$, formuła $f_9 \in S_1$ jest równoważna z podcelem p_7). Podcel p_7 należy do zbioru $P = \{p_5, p_6, p_7, p_8, p_9, p_{10}\}$ podcelów koniunkcyjnych, utworzonych z p_2 , za pomocą reguły przekształcania O_2 . Po podstawieniu w zbiorze P : $x_1 := SKRYNKA$, otrzymujemy:

$$P = \{ (p_5) TYP(SKRYNKA, BOX), \\ (p_6) NA PODŁODZE(ROBOT), \\ (p_7) PRZY(ROBOT, SKRYNKA), \\ (p_8) W(SKRYNKA, x_2), \\ (p_9) W(WYŁĄCZNIK, x_2), \\ (p_{10}) DO PODNIESIENIA(SKRYNKA) \}.$$

Przyporządkowania:

$$(p_5) \longleftrightarrow (f_7), (p_6) \longleftrightarrow (f_4), \\ (p_7) \longleftrightarrow (f_9), (p_8) \longleftrightarrow (f_2), \\ (p_9) \longleftrightarrow (f_3), (p_{10}) \longleftrightarrow (f_5)$$

pozwalają wydzielić zbiór podstawień X_2^*

$$X_2^* = \{ x_1 := SKRYNKA, x_2 := POKÓJ, x_3 := POKÓJ \}$$

i przekształcić S_1 w S_2 , za pomocą reguły przekształcania O_2 (jest to drugi krok pionowy).

$$S_2 = \{ (f_1) \text{ W(ROBOT, POKÓJ)}, \\ (f_2) \text{ W(SKRYNKA, POKÓJ)}, \\ (f_3) \text{ W(WYŁĄCZNIK, POKÓJ)}, \\ (f_4) \text{ NA PODŁODZE(ROBOT)}, \\ (f_5) \text{ DO PODNIESIENIA(SKRYNKA)}, \\ (f_6) \text{ STAN(WYŁĄCZNIK, WYŁĄCZONY)}, \\ (f_7) \text{ TYP(SKRYNKA, BOX)}, \\ (f_8) \text{ TYP(WYŁĄCZNIK, SWITCH)}, \\ (f_{10}) \text{ PRZY(SKRYNKA, WYŁĄCZNIK)}, \\ (f_{11}) \text{ PRZY(ROBOT, SKRYNKA)} \}.$$

Utworzona sytuacja S_2 zawiera cel p_2 (po podstawieniu w p_2 : $x_1 := \text{SKRYNKA}$, formuła $f_1 \in S_2$ jest równoważna z podcelem p_2). Podcel p_2 należy do zbioru $P = \{p_2, p_3, p_4\}$ podcelów koniunkcyjnych, utworzonych z p_1 , za pomocą reguły przekształcania O_4 . Po podstawieniu w zbiorze P : $x_1 := \text{SKRYNKA}$, otrzymujemy:

$$P = \{ (p_2) \text{ PRZY(ROBOT, WYŁĄCZNIK)}, \\ (p_3) \text{ TYP(WYŁĄCZNIK, SWITCH)}, \\ (p_4) \text{ NA(ROBOT, SKRYNKA)} \}.$$

Tylko podcele p_2 i p_3 z P można przyporządkować formułom S_2 :

$$(p_2) \longleftrightarrow (f_{10}), \quad (p_3) \longleftrightarrow (f_8),$$

natomiast podcel p_4 nie może być przyporządkowany żadnej formule z S_2 . Stąd:

$$P^+ = \{p_2, p_3\}, \quad P^- = \{p_4\}.$$

Nie można wykonać kroku pionowego, istnieje natomiast możliwość wykonania kroku poziomego, gdyż p_4 posiada w DP rozbięcie na podcele p_{11} , p_{12} , p_{13} , które można przyporządkować formułom S_2 :

$$\begin{aligned}(p_{11}) &\longleftrightarrow (f_9), (p_{12}) \longleftrightarrow (f_4), \\ (p_{13}) &\longleftrightarrow (f_7).\end{aligned}$$

Zbiór podstawień $X_3^* = X_2^*$. Sytuacja S_2 , przekształcona za pomocą reguły O_3 , daje sytuację S_3 :

$$\begin{aligned}S_3 = \{ &(f_1) \text{ W(ROBOT, POKÓJ)}, \\ &(f_2) \text{ W(SKRZYNKA, POKÓJ)}, \\ &(f_3) \text{ W(WYŁĄCZNIK, POKÓJ)}, \\ &(f_5) \text{ DO PODNIESIENIA(SKRZYNKA)}, \\ &(f_6) \text{ STAN(WYŁĄCZNIK, WYŁĄCZONY)}, \\ &(f_7) \text{ TYP(SKRZYNKA, BOX)}, \\ &(f_8) \text{ TYP(WYŁĄCZNIK, SWITCH)}, \\ &(f_{10}) \text{ PRZY(SKRZYNKA, WYŁĄCZNIK)}, \\ &(f_{11}) \text{ PRZY(ROBOT, SKRZYNKA)}, \\ &(f_{12}) \text{ NA(ROBOT, SKRZYNKA)}\}.\end{aligned}$$

Utworzona sytuacja S_3 zawiera cel p_4 (po podstawieniu w p_4 : $x_1 := \text{SKRZYNKA}$, formuła $f_{12} \in S_3$ jest równoważna z podcelem p_4), zawiera również podcele p_2 i p_3 , koniunkcyjne względem p_4 . Zaisztniały zatem w S_3 warunki do wykonania kroku pionowego. Przyporządkowania:

$$\begin{aligned}(p_2) &\longleftrightarrow (f_{10}), (p_3) \longleftrightarrow (f_8), \\ (p_4) &\longleftrightarrow (f_{12}),\end{aligned}$$

ze zbiorem podstawień $X_4^* = X_3^*$, pozwalają przekształcić sytuację S_3 , za pomocą reguły przekształcania O_4 , w nową sytuację S_4 :

$$\begin{aligned}S_4 = \{ &(f_1) \text{ W(ROBOT, POKÓJ)}, \\ &(f_2) \text{ W(SKRZYNKA, POKÓJ)}, \\ &(f_3) \text{ W(WYŁĄCZNIK, POKÓJ)}, \\ &(f_5) \text{ DO PODNIESIENIA(SKRZYNKA)}, \\ &(f_7) \text{ TYP(SKRZYNKA, BOX)}, \\ &(f_8) \text{ TYP(WYŁĄCZNIK, SWITCH)},\end{aligned}$$

(f_{10}) PRZY(SKRZYNKA,WYŁĄCZNIK),
 (f_{11}) PRZY(ROBOT,SKRZYNKA),
 (f_{12}) NA(ROBOT,SKRZYNKA),
 (f_{13}) STAN(WYŁĄCZNIK,ZAŁĄCZONY)}.

Utworzona sytuacja S_4 zawiera cel główny p_1 w DP, więc analiza zakończyła się sukcesem. Ciąg przekształceń:

$$s_p \xrightarrow{O_1} s_{11} \xrightarrow{O_2} s_2 \xrightarrow{O_3} s_3 \xrightarrow{O_4} s_4, s_k \subset s_4,$$

stanowi poszukiwaną sekwencję przekształcającą $\mathcal{T}$ - rozwiązanie zadania, sformułowanego w p. 3.3.

5. PROPOZYCJA METODY POSZUKIWANIA SEKWENCJI PRZEKSZTAŁCAJĄCEJ W PRZESTRZENI STANÓW

Omówione w rozdziale 4. podstawowe akcje wykorzystywane są w metodzie poszukiwania sekwencji przekształcającej w przestrzeni stanów. Metoda ta będzie przedstawiona w niniejszym rozdziale w trzech wariantach, odpowiadających trzem różnym strategiom. Każda z tych strategii stanowi ogólną i stosowaną często w systemach poszukiwania rozwiązań metodykę rozwoju przestrzeni poszukiwania rozwiązania, stąd ich użycie w poszczególnych wariantach zbudowanej metody.

Poszczególnymi wariantami metody są:

- wariant A, z użyciem strategii rozwoju "wszerz",
- wariant B, z użyciem strategii rozwoju "w głąb",
- wariant C, z użyciem strategii uporządkowanego wyboru,
z dwiema funkcjami ocen.

Ponadto zostanie przedstawiony algorytm automatycznej modyfikacji funkcji oceny, będący mechanizmem adaptacyjnym metody.

5.1. Opis wariantu A metody - strategia rozwoju "wszerz"

Poniżej przedstawiono poszczególne kroki wariantu A metody oraz związki między nimi. Zakłada się stale, że jeśli warunek, sprawdzany w danym kroku metody, nie jest spełniony lub jeśli polecona czynność jest niewykonalna, to następuje przejście do dalszej części opisu. Użyty w opisie termin "lista" określa uporządkowany zbiór opisów odpowiednich obiektów, występujących w metodzie.

- ① Sprawdzić, czy $S_k \subseteq S_p$. Jeśli tak - stwierdzono istnienie celu w sytuacji początkowej (błąd w sformułowaniu zadania) - przejść do ⑮.

- ② Utworzyć listę 0 kolejnych reguł przekształcania (kolejność reguł określona arbitralnie). Założyć ograniczenia ilościowe (tzw. limity) dla metody, w postaci maksymalnych ilości:
- $n_{\max}$ wierzchołków w DP oraz $m_{\max}$ wierzchołków w DR
(są to równocześnie limity rozmiarów przestrzeni poszukiwania rozwiązania),
 - $k_{\max}$ przebiegów analizy DP (przebiegiem analizy DP jest ciąg akcji od rozpoczęcia analizy do jej niepowodzenia lub sukcesu),
- które wystąpią podczas poszukiwania sekwencji przekształcającej $\mathcal{T}$.
- ③ Na podstawie opisu celu S_k utworzyć zbiór koniunkcyjny wierzchołków początkowych DP. Utworzyć listę W z tymi wierzchołkami. Utworzyć wierzchołek początkowy S_p w DR, na podstawie opisu sytuacji początkowej S_p . Utworzyć listę S z tym wierzchołkiem.
- ④ Wybrać wierzchołek p w DP, będący pierwszym elementem listy W, usuwając go jednocześnie z W.
- ⑤ Wybrać z listy 0 pierwszą regułę O_1 , umożliwiającą rozbudowanie DP od wierzchołka p. Utworzyć puste listy WN i SN.
- ⑥ Jeśli w liście 0 brak takiej reguły, przejść do ⑬.
- Dla wierzchołka p oraz reguły O_1 utworzyć zbiór wektorów podstawień: $\{V_1^{*1}, V_1^{*2}, \dots, V_1^{*r}\}$, $r \geq 1$. Utworzyć listę VN z kolejnymi wektorami podstawień.
- ⑦ Wybrać z listy VN pierwszy wektor podstawienia V_1^* , usuwając go jednocześnie z VN. Rozbudować DP od wierzchołka p, za pomocą reguły O_1 i wektora V_1^* . Dopisać do listy WN zbiór wierzchołków DP, powstałych podczas rozbudowy DP w tym kroku.
- Jeśli podczas rozbudowy DP osiągnięto limit $n_{\max}$ - przejść

do (16).

- (8) Wybrać wierzchołek S_i , będący pierwszym elementem listy S .
- (9) Przeprowadzić analizę DP od pierwszego z ostatnio dopisanych do listy WN wierzchołków, względem wierzchołka S_i (p. 4.3). Dopisywać do listy SN wierzchołki DR, powstające w wyniku analizy DP.
Jeśli podczas budowy DR osiągnięto limit $m_{\max}$ - przejść do (16).
Jeśli analiza DP doprowadziła do sukcesu, czyli do osiągnięcia celu głównego w DP - przejść do (17).
- (10) Jeśli osiągnięto limit $k_{\max}$ - przejść do (16).
- (11) Jeśli wierzchołek S_i nie był ostatnim elementem listy S , wybrać następny element listy S jako wierzchołek S_i i przejść do (9).
Jeśli lista VN nie jest pusta - przejść do (7).
- (12) Dołączyć listę SN do końca listy S , usuwając jednocześnie z SN wszystkie jej elementy. Wybrać z listy O kolejną regułę O_1 , umożliwiającą rozbudowanie DP od wierzchołka p. Przejść do (6).
- (13) Dołączyć listę WN do końca listy W .
- (14) Jeśli lista W nie jest pusta - przejść do (4).
- (15) Sekwencja przekształcająca $\mathcal{T}$ nie istnieje. Przejść do (18).
- (16) Nie znaleziono sekwencji przekształcającej $\mathcal{T}$ dla zadanych limitów. Przejść do (18).
- (17) Odczytać sekwencję przekształcającą $\mathcal{T}$ z DR, jako ścieżkę od wierzchołka początkowego S_p do wierzchołka, utworzonego w wyniku osiągnięcia celu głównego w DP.
- (18) Koniec działania metody.

Przedstawiony wariant metody, pozbawiony limitów $n_{\max}$, $m_{\max}$ i $k_{\max}$, umożliwia teoretycznie znalezienie sekwencji przekształcającej $\mathcal{T}$ zawsze, gdy ona istnieje dla danego zadania. W wariacie tym DP rozwija się "wszerz", podobnie jak DR. Analizowane są wszystkie zapowiadające sukces warianty ścieżek w DP, względem każdego wierzchołka DR, czyli względem każdej sytuacji, powstałej podczas poprzednich przebiegów analizy DP. Analiza DP każdorazowo rozpoczyna się względem wierzchołka początkowego S_p w DR, stąd ewentualnie uzyskana sekwencja przekształcająca $\mathcal{T}$ zawiera najmniejszą z możliwych liczbę przekształceń.

PRZYKŁAD. Działanie wariantu A metody zilustrujemy rozwiązaniem zadania, sformułowanego w p. 3.3.

$$(1) S_k \not\subseteq S_d$$

$$(2) O := \{O_1, O_2, O_3, O_4\}$$

Dla uproszczenia przykładu założymy istnienie dostatecznie dużych limitów $n_{\max}$, $m_{\max}$ i $k_{\max}$, by pominąć sprawdzanie, czy nie zostały one osiągnięte.

$$(3) W := \{p_1\}, (p_1) \text{ STAN(WYŁĄCZNIK, ZAŁĄCZONY)} \\ S := \{S_p\}$$

$$(4) p := p_1, W := W \setminus \{p_1\} \Rightarrow W = \emptyset$$

$$(5) O_1 := O_4, WN := \emptyset, SN := \emptyset$$

$$(6) V_4^{*1} = (x_1, \text{WYŁĄCZNIK}), VN := \{V_4^{*1}\}$$

$$(7) V_1^{*1} := V_4^{*1}, VN := VN \setminus \{V_4^{*1}\} \Rightarrow VN = \emptyset$$

$$(p_2) \text{ PRZY}(x_1, \text{WYŁĄCZNIK})$$

$$(p_3) \text{ TYP(WYŁĄCZNIK, SWITCH)}$$

$$(p_4) \text{ NA(ROBOT, } x_1)$$

$$WN := WN \cup \{p_2, p_3, p_4\} \Rightarrow WN = \{p_2, p_3, p_4\}$$

$$(8) S_1 := S_p$$

$$(9) \text{ Następuje niepowodzenie analizy DP w pierwszym kroku,}$$

ponieważ nie można przyporządkować podcelów p_2 i p_4 formułom sytuacji S_i .

$$SN = \emptyset$$

$$(10) \rightarrow (11) \rightarrow (12)$$

$$(12) \quad S := S \cup SN \Rightarrow S = \{S_p\}, \quad SN = \emptyset$$

$$(6) \rightarrow (13)$$

$$(13) \quad W := W \cup WN \Rightarrow W = \{p_2, p_3, p_4\}$$

$$(14) \rightarrow (4)$$

$$(4) \quad p := p_2, \quad W := W \setminus \{p_2\} \Rightarrow W = \{p_3, p_4\}$$

$$(5) \quad O_1 := O_1, \quad WN = \emptyset$$

$$(6) \quad v_1^{*1} = (x_2, \text{WYŁĄCZNIK}, q_3), \quad VN := \{v_1^{*1}\}$$

$$(7) \quad v_1^* := v_1^{*1}, \quad VN := VN \setminus \{v_1^{*1}\} \Rightarrow VN = \emptyset$$

(p_5) NA PODŁODZE(ROBOT)

(p_6) W(ROBOT, x_2)

(p_7) W(WYŁĄCZNIK, x_2)

$$WN := WN \cup \{p_5, p_6, p_7\} \Rightarrow WN = \{p_5, p_6, p_7\}$$

$$(8) \quad S_i := S_p$$

$$(9) \quad \left. \begin{array}{l} (p_5) \leftrightarrow f_4 \in S_p \\ (p_6) \leftrightarrow f_1 \in S_p \\ (p_7) \leftrightarrow f_3 \in S_p \end{array} \right\} \Rightarrow X_{11}^* = \{x_2 := \text{POKÓJ}\}$$

$$O_{11}: S_p \xrightarrow{01} S_1$$

$$S_{11} = \{(f_1) \text{ W(ROBOT, POKÓJ)},$$

$(f_2) \text{ W(SKRYNKA, POKÓJ)},$

$(f_3) \text{ W(WYŁĄCZNIK, POKÓJ)},$

$(f_4) \text{ NA PODŁODZE(ROBOT)},$

$(f_5) \text{ DO PODNIESIENIA(SKRYNKA)},$

$(f_6) \text{ STAN(WYŁĄCZNIK, WYŁĄCZONY)},$

$(f_7) \text{ TYP(SKRYNKA, BOX)},$

$(f_8) \text{ TYP(WYŁĄCZNIK, SWITCH)},$

$(f_9) \text{ PRZY(ROBOT, WYŁĄCZNIK)}\}$

$$S_1 \leftrightarrow p_2, \quad p_2 \in \{p_2, p_3, p_4\}$$

Następuje niepowodzenie analizy DP w drugim kroku, ponieważ podcelowi p_4 nie można przyporządkować żadnej z formuł sytuacji S_1 , oraz nie jest możliwe osiągnięcie podcelu p_4 z sytuacji S_1 .

$$SN = \{S_1\}$$

$$\textcircled{10} \rightarrow \textcircled{11} \rightarrow \textcircled{12}$$

$$\textcircled{12} \quad S := S \cup SN \Rightarrow S = \{S_p, S_1\}, \quad SN = \emptyset$$

$$O_1 := O_2$$

$$\textcircled{6} \quad V_2^{*1} = (x_1, \text{WYŁĄCZNIK}, x_3), \quad V_2^{*2} = (\text{WYŁĄCZNIK}, x_5, x_4),$$

$$VN := \{V_2^{*1}, V_2^{*2}\}$$

$$\textcircled{7} \quad V_1^{*1} := V_2^{*1}, \quad VN := VN \setminus \{V_2^{*1}\} \Rightarrow VN = \{V_2^{*2}\}$$

$$(p_8) \quad \text{TYP}(x_1, \text{BOX})$$

$$(p_9) \quad \text{NA. PODŁODZE}(\text{ROBOT})$$

$$(p_{10}) \quad \text{PRZY}(\text{ROBOT}, x_1)$$

$$(p_{11}) \quad \text{W}(x_1, x_3)$$

$$(p_{12}) \quad \text{W}(\text{WYŁĄCZNIK}, x_3)$$

$$(p_{13}) \quad \text{DO PODNIESIENIA}(x_1)$$

$$WN := WN \cup \{p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}\} \Rightarrow WN = \{p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}\}$$

$$\textcircled{8} \quad S_i := S_p$$

$\textcircled{9}$ Następuje niepowodzenie analizy DP w pierwszym kroku, ponieważ nie można przyporządkować podcelu p_{10} żadnej formule sytuacji S_p .

$$SN = \emptyset$$

$$\textcircled{10} \rightarrow \textcircled{11}$$

$$\textcircled{11} \quad S_i := S_{1i}$$

$\textcircled{9}$ Następuje niepowodzenie analizy DP w pierwszym kroku, ponieważ nie można przyporządkować podcelów: p_8 , p_{10} i p_{13} formułom sytuacji S_1 .

$$SN = \emptyset$$

$$\textcircled{10} \rightarrow \textcircled{11} \rightarrow \textcircled{7}$$

$$(7) \quad V_1^* := V_2^{*2}, \quad VN := VN \setminus \{V_2^{*2}\} \Rightarrow VN = \emptyset$$

(p₁₄) TYP(WYŁĄCZNIK, BOX)

(p₁₅) NA PODŁODZE(ROBOT)

(p₁₆) PRZY(ROBOT, WYŁĄCZNIK)

(p₁₇) W(WYŁĄCZNIK, x₄)

(p₁₈) W(x₅, x₄)

(p₁₉) DO PODNIESIENIA(WYŁĄCZNIK)

$$WN := WN \cup \{p_{14}, p_{15}, p_{16}, p_{17}, p_{18}, p_{19}\} \Rightarrow WN = \{p_5, p_6, \dots, p_{19}\}$$

$$(8) \quad S_i := S_p$$

(9) Następuje niepowodzenie analizy DP w pierwszym kroku, ponieważ nie można przyporządkować podcelów: p₁₄, p₁₆ i p₁₉ formułom sytuacji S_p.

$$SN = \emptyset$$

$$(10) \rightarrow (11)$$

$$(11) \quad S_i := S_{11}$$

(9) Następuje niepowodzenie analizy DP w pierwszym kroku, ponieważ nie można przyporządkować podcelów: p₁₄ i p₁₉ formułom sytuacji S₁.

$$SN = \emptyset$$

$$(10) \rightarrow (11) \rightarrow (12)$$

$$(12) \quad S := S \cup SN \Rightarrow S = \{S_p, S_1\}, \quad SN = \emptyset$$

$$(6) \rightarrow (13)$$

$$(13) \quad W := W \cup WN \Rightarrow W = \{p_3, p_4, \dots, p_{19}\}$$

$$(14) \rightarrow (4)$$

$$(4) \quad p := p_3, \quad W := W \setminus \{p_3\} \Rightarrow W = \{p_4, p_5, \dots, p_{19}\}$$

$$(5) \quad WN = \emptyset$$

$$(6) \rightarrow (13)$$

$$(13) \quad W := W \cup WN = W = \{p_4, p_5, \dots, p_{19}\}$$

$$(14) \rightarrow (4)$$

$$(4) \quad p := p_4, \quad W := W \setminus \{p_4\} \Rightarrow W = \{p_5, p_6, \dots, p_{19}\}$$

$$(5) \quad O_1 := O_3, \quad WN = \emptyset$$

$$\textcircled{6} \quad V_3^{*1} = (x_1), \quad VN := \{V_3^{*1}\}$$

$$\textcircled{7} \quad V_1^* := V_3^{*1}, \quad VN := VN \setminus \{V_3^{*1}\} \Rightarrow VN = \emptyset$$

$$(p_{20}) \text{ PRZY}(\text{ROBOT}, x_1)$$

$$(p_{21}) \text{ NA PODŁODZE}(\text{ROBOT})$$

$$(p_{22}) \text{ TYP}(x_1, \text{BOX})$$

$$WN := WN \cup \{p_{20}, p_{21}, p_{22}\} \Rightarrow WN = \{p_{20}, p_{21}, p_{22}\}$$

$$\textcircled{8} \quad S_i := S_p$$

$\textcircled{9}$ Następuje niepowodzenie analizy DP w pierwszym kroku, ponieważ podcelowi p_{20} nie można przyporządkować żadnej z formuł sytuacji S_p .

$$SN = \emptyset$$

$$\textcircled{10} \rightarrow \textcircled{11}$$

$$\textcircled{11} \quad S_i := S_{1i}$$

$\textcircled{9}$ Następuje niepowodzenie analizy DP w pierwszym kroku, ponieważ podcelom: p_{20} i p_{22} nie można przyporządkować żadnej z formuł sytuacji S_1 .

$$SN = 0$$

$$\textcircled{10} \rightarrow \textcircled{11} \rightarrow \textcircled{12}$$

$$\textcircled{12} \quad S := S \cup SN \Rightarrow S = \{S_p, S_1\}, \quad SN = \emptyset$$

$$\textcircled{6} \rightarrow \textcircled{13}$$

$$\textcircled{13} \quad W := W \cup WN \Rightarrow W = \{p_5, p_6, \dots, p_{22}\}$$

$$\textcircled{14} \rightarrow \textcircled{4}$$

⋮

Kolejne próby rozbudowy DP od wierzchołków: p_5, p_6, p_7, p_8 i p_9 są nieskuteczne, ponieważ w liście 0 brak reguł przekształcania, umożliwiających rozbudowę DP. Zmiana tego stanu następuje dopiero wówczas, gdy:

$$\textcircled{4} \quad p := p_{10}, \quad W := W \setminus \{p_{10}\} \Rightarrow W = \{p_{11}, p_{12}, \dots, p_{22}\}$$

$$\textcircled{5} \quad O_1 := O_1, \quad WN = \emptyset$$

$$\textcircled{6} \quad V_1^{*1} = (x_6, x_1, x_7), \quad VN := \{V_1^{*1}\}$$

$$\textcircled{7} \quad V_1^* := V_1^{*1}, \quad VN := VN \setminus \{V_1^{*1}\} \Rightarrow VN = \emptyset$$

(p₂₃) NA PODŁODZE(ROBOT)

(p₂₄) W(ROBOT, x₆)

(p₂₅) W(x₁, x₆)

$$WN := WN \setminus \{p_{23}, p_{24}, p_{25}\} \Rightarrow WN = \{p_{23}, p_{24}, p_{25}\}$$

(Utworzone do tego kroku włącznie DP przedstawiono na rys.

5. 1.)

$$\textcircled{8} \quad S_i := S_p$$

$$\textcircled{9} \quad \left. \begin{array}{l} (p_{23}) \longleftrightarrow f_4 \in S_p \\ (p_{24}) \longleftrightarrow f_1 \in S_p \\ (p_{25}) \longleftrightarrow f_2 \in S_p \end{array} \right\} \Rightarrow X_2^* = \{x_1 := \text{SKRZYŃKA}, x_2 := \text{POKÓJ}\}$$

$$O_1: S_p \xrightarrow{0_1} S_2$$

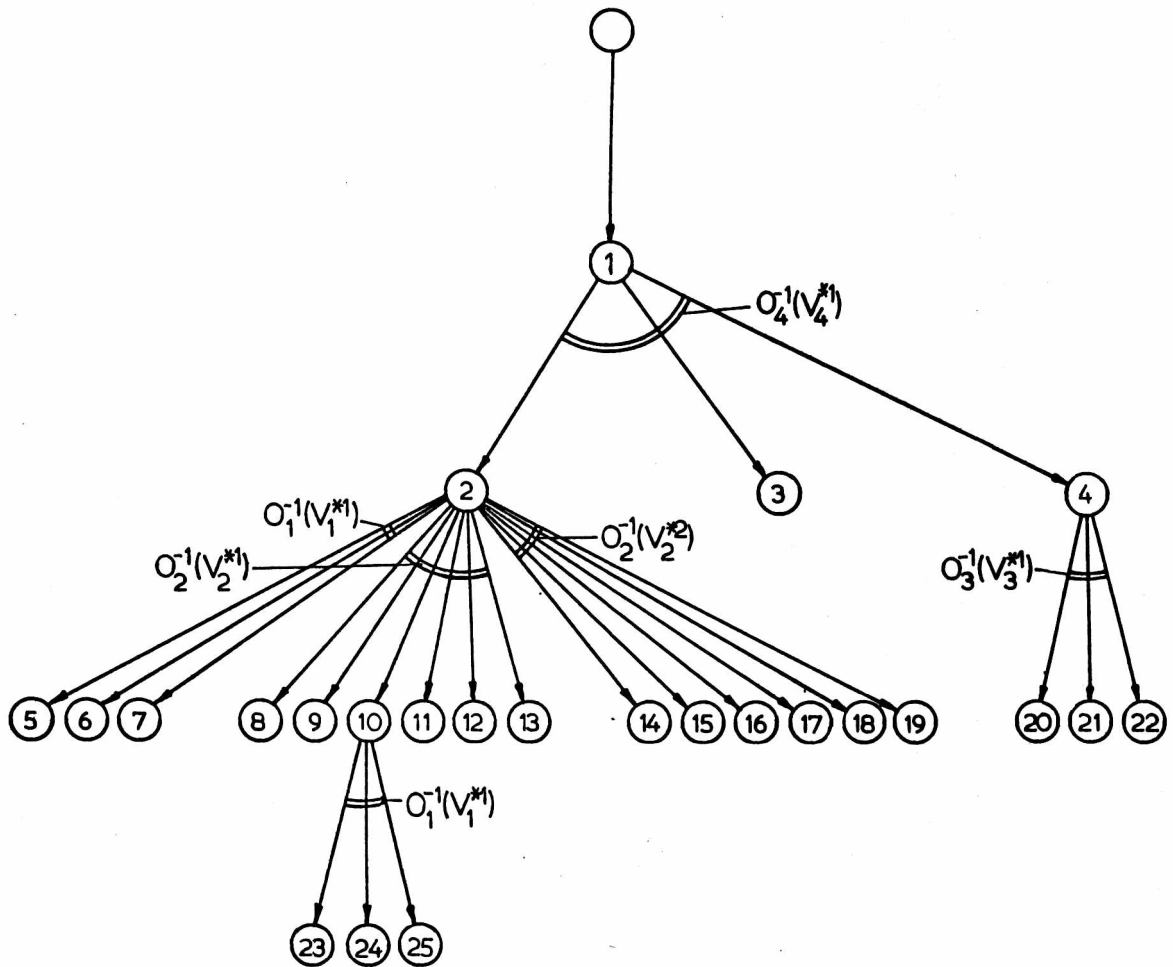
$$S_2 = \{ (f_1) \text{ W(ROBOT, POKÓJ)}, \\ (f_2) \text{ W(SKRYŃKA, POKÓJ)}, \\ (f_3) \text{ W(WYŁĄCZNIK, POKÓJ)}, \\ (f_4) \text{ NA PODŁODZE(ROBOT)}, \\ (f_5) \text{ DO PODNIESIENIA(SKRYŃKA)}, \\ (f_6) \text{ STAN(WYŁĄCZNIK, WYŁĄCZONY)}, \\ (f_7) \text{ TYP(SKRYŃKA, BOX)}, \\ (f_8) \text{ TYP(WYŁĄCZNIK, SWITCH)}, \\ (f_{10}) \text{ PRZY(ROBOT, SKRYŃKA)} \}$$

$$S_2 \longleftrightarrow p_{10}, p_{10} \in \{p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}\}, SN = \{S_2\}$$

$$\left. \begin{array}{l} (p_8) \longleftrightarrow f_7 \in S_2 \\ (p_9) \longleftrightarrow f_4 \in S_2 \\ (p_{10}) \longleftrightarrow f_{10} \in S_2 \\ (p_{11}) \longleftrightarrow f_2 \in S_2 \\ (p_{12}) \longleftrightarrow f_3 \in S_2 \\ (p_{13}) \longleftrightarrow f_5 \in S_2 \end{array} \right\} \Rightarrow X_3^* = \{x_1 := \text{SKRZYŃKA}, x_3 := \text{POKÓJ}, x_6 := \text{POKÓJ}\}$$

$$O_2: S_2 \xrightarrow{0_2} S_3$$

$$S_3 = \left\{ \begin{array}{l} (f_1) \\ \vdots \\ (f_8) \end{array} \right\} \text{ jak w } S_2$$



- | | |
|--|--|
| (p ₁) STAN(WYŁĄCZNIK, ZAŁĄCZONY) | (p ₁₄) TYP(WYŁĄCZNIK, BOX) |
| (p ₂) PRZY(x ₁ , WYŁĄCZNIK) | (p ₁₅) NA PODŁODZE(ROBOT) |
| (p ₃) TYP(WYŁĄCZNIK, SWITCH) | (p ₁₆) PRZY(ROBOT, WYŁĄCZNIK) |
| (p ₄) NA(ROBOT, x ₁) | (p ₁₇) W(WYŁĄCZNIK, x ₄) |
| (p ₅) NA PODŁODZE(ROBOT) | (p ₁₈) W(x ₅ , x ₄) |
| (p ₆) W(ROBOT, x ₂) | (p ₁₉) DO PODNIESIENIA(WYŁĄCZNIK) |
| (p ₇) W(WYŁĄCZNIK, x ₂) | |
| (p ₈) TYP(x ₁ , BOX) | (p ₂₀) PRZY(ROBOT, x ₁) |
| (p ₉) NA PODŁODZE(ROBOT) | (p ₂₁) NA PODŁODZE(ROBOT) |
| (p ₁₀) PRZY(ROBOT, x ₁) | (p ₂₂) TYP(x ₁ , BOX) |
| (p ₁₁) W(x ₁ , x ₃) | (p ₂₃) NA PODŁODZE(ROBOT) |
| (p ₁₂) W(WYŁĄCZNIK, x ₃) | (p ₂₄) W(ROBOT, x ₆) |
| (p ₁₃) DO PODNIESIENIA(x ₁) | (p ₂₅) W(x ₁ , x ₆) |

Rys. 5.1. Drzewo podcelów DP dla zadania z przykładu w p. 5.1

$(f_{11}) \text{ PRZY}(\text{SKRZYŃKA}, \text{WYŁĄCZNIK}),$

$(f_{12}) \text{ PRZY}(\text{ROBOT}, \text{SKRZYŃKA})\}$

$S_3 \longleftrightarrow p_2, p_2 \in \{p_2, p_3, p_4\}, SN = \{S_2, S_3\}$

$(p_2) \longleftrightarrow f_{11} \in S_3$

$(p_3) \longleftrightarrow f_8 \in S_3$

Dla podcelu p_4 brak przyporządkowania jw., lecz podcel ten posiada swoje rozwinięcie w DP, w postaci podcelów:

$\{p_{20}, p_{21}, p_{22}\}.$

$(p_{20}) \longleftrightarrow f_{12} \in S_3$

$(p_{21}) \longleftrightarrow f_4 \in S_3$

$(p_{22}) \longleftrightarrow f_7 \in S_3$

$O_3: S_3 \xrightarrow{O_3} S_4$

$S_4 = \left\{ \begin{array}{l} (f_{11}) \\ \vdots \\ (f_3) \\ (f_5) \\ \vdots \\ (f_8) \\ (f_{11}) \\ (f_{12}) \end{array} \right\}$ jak w S_3 , za wyjątkiem $f_4 \in S_3$

$(f_{13}) \text{ NA}(\text{ROBOT}, \text{SKRZYŃKA})\}$

$S_4 \longleftrightarrow p_4, p_4 \in \{p_2, p_3, p_4\}, SN = \{S_2, S_3, S_4\}$

$(p_2) \longleftrightarrow f_{11} \in S_4$

$(p_3) \longleftrightarrow f_8 \in S_4$

$(p_4) \longleftrightarrow f_{13} \in S_4$

$O_4: S_4 \xrightarrow{O_4} S_5$

$S_5 = \left\{ \begin{array}{l} (f_1) \\ (f_2) \\ (f_3) \\ (f_5) \\ (f_7) \\ (f_8) \end{array} \right\}$ jak w S_4 , za wyjątkiem $f_6 \in S_4$

$$\left. \begin{array}{l} (f_{11}) \\ (f_{12}) \\ (f_{13}) \end{array} \right\} \\ (f_{14}) \text{ STAN(WYŁĄCZNIK,ZAŁĄCZONY)}$$

$s_5 \longleftrightarrow p_1$, osiągnięto cel główny w DP

$$(17) \quad \mathcal{T}: s_p \xrightarrow{0_1} s_2 \xrightarrow{0_2} s_3 \xrightarrow{0_3} s_4 \xrightarrow{0_4} s_5$$

(18) Koniec

5.2. Opis wariantu B metody - strategia rozwoju "w głąb"

Gdy możliwe jest określenie maksymalnej liczby przekształceń w poszukiwanej sekwencji przekształcającej $\mathcal{T}$, wówczas można zastosować strategię rozwoju DP "w głąb", rozbudowując DP na zadaną "głębokość" graniczną $g_{\max}$ w sposób systematyczny. Opis kroków takiej metody będzie różnił się od opisu wariantu A jedynie postacią kroków 4. i 13. - po uzupełnieniu kroku 2. o limit $g_{\max}$, dlatego wariant B metody przedstawimy w oparciu o opis wariantu A, zamieszczony w p. 5.1.

W wariantcie B, stosującym strategię rozwoju "w głąb", krok 4. ma postać:

- (4) Wybrać wierzchołek p w DP, będący pierwszym elementem listy W , usuwając go równocześnie z W . Jeśli poziom wierzchołka p w DP osiągnął $g_{\max}$ - przejść do (14).

Natomiast krok 13. ma postać:

- (13) Dołączyć listę WW do początku listy W .

Podobnie, jak to ma miejsce w wariantcie A, również wariant B, pozbawiony limitów $n_{\max}$, $m_{\max}$ i $k_{\max}$, umożliwia teoretycznie znalezienie sekwencji przekształcającej $\mathcal{T}$ zawsze, gdy ona istnieje dla danego zadania. W wariantcie tym DP rozwija się "w głąb", do poziomu granicznego $g_{\max}$.

5.3. Opis wariantu C metody - strategia uporządkowanego wyboru

Dla realizacji strategii uporządkowanego wyboru w metodzie dwukierunkowego poszukiwania sekwencji przekształcającej $\mathcal{T}$ określono dwie funkcje ocen:

- funkcję oceny dla wierzchołków w drzewie podcelów,
- funkcję oceny dla wierzchołków w drzewie rozwiązania.

W ogólnym przypadku postaci funkcji ocen dostosowywane są do konkretnych klas zadań. Przedstawione poniżej postacie funkcji orientują metodę na zastosowania do rozwiązywania klasycznych zadań, testujących metody poszukiwania rozwiązań, a więc na zadania ogólne.

5.3.1. Funkcja oceny dla wierzchołków w drzewie podcelów

Dla oceny wierzchołka (podcelu) p w drzewie podcelów DP, od którego należy rozwijać DP, wprowadzono funkcję g za pomocą podanych niżej funkcji składowych.

1. Funkcja

$$g_1(v_p) = \frac{1}{1 + v_p},$$

gdzie v_p określa numer poziomu w DP, na którym znajduje się wierzchołek p . Funkcja ta ma różnicować wierzchołki na różnych poziomach DP (g_1 maleje wraz ze wzrostem numeru poziomu). Wierzchołki, posiadające większą wartość funkcji g_1 , leżą bliżej korzenia DP, a więc odpowiadające im podcele leżą bliżej celu głównego.

2. Funkcja

$$g_2(f_p, k_p) = \frac{f_p}{k_p},$$

gdzie: f_p określa wartość funkcji g_2 , wyznaczoną wcześniej

dla podcelu nadrzędnego względem p (przyjmuje się, że korzeń DP ma wartość funkcji g_2 równą 1), k_p określa ilość podcelów koniunkcyjnych w zbiorze, do którego należy podcel p . Funkcja g_2 ma różnicować wierzchołki, leżące w częściach drzewa DP o różnym stopniu rozgałęzienia (g_2 maleje wraz ze wzrostem rozgałęzienia). Wierzchołki z większą wartością g_2 dają większe szanse powodzenia analizie DP, ze względu na mniejszą ilość podcelów na drodze do celu głównego.

3. Funkcja

$$g_3(m_p, m_c) = \frac{m_p}{1 + m_c},$$

gdzie: m_p określa ilość dotychczasowych analiz DP, zakończonych niepowodzeniem na wierzchołku p , m_c określa łączną ilość niepowodzeń analiz DP. Funkcja ta różnicuje podcele pod względem trudności w ich osiągnięciu (g_3 rośnie wraz ze wzrostem ilości analiz DP, które zakończyły się niepowodzeniem na danym podcelu). Wierzchołki z większą wartością funkcji g_3 , jako reprezentujące trudniej osiągalne podcele, powinny być w pierwszej kolejności rozbite na łatwiej osiągalne podcele składowe.

4. Funkcja

$$g_4(t_p, t_o) = \frac{t_p}{t_o},$$

gdzie: t_o określa ilość reguł przekształcania w zbiorze O , t_p określa ilość reguł przekształcania w podzbiorze O_p reguł, za pomocą których można rozbudować DP od wierzchołka p . Funkcja ta różnicuje podcele pod względem łatwości ich kolejnego rozbijania na nowe podcele, za pomocą reguł przekształcania. Rozbudowanie DP od wierzchołków z większą wartością funkcji g_4 daje szansę przeanalizowania większej

liczby alternatywnych wariantów, bez zbędnej rozbudowy DP w różnych kierunkach.

5. Funkcja

$$g_5(l_{Ad}, l_o) = \begin{cases} 0 & , \text{ gdy } O_p = \emptyset \\ \frac{l_{Ad}}{l_o} & , \text{ gdy } O_p \neq \emptyset \end{cases}$$

gdzie: l_o określa łączną ilość formuł w opisach reguł przekształcania w podzbiorze O_p reguł, za pomocą których można rozbudować DP od wierzchołka p , l_{Ad} określa łączną ilość formuł w zbiorach Ad reguł przekształcania z podzbioru O_p . Funkcja różnicuje podcele pod względem ilości opisów nowych stanów, wprowadzanych do przestrzeni stanów $\mathcal{P}$ przez reguły przekształcania, użyte do ewentualnego rozbicia danego podcelu p . Bardziej celowe jest rozbudowywanie DP od wierzchołków, posiadających w podzbiorze O_p reguły przekształcania z większą ilością formuł Ad , ponieważ każda formuła z Ad w regule, użytej do przekształcania sytuacji, wprowadza nowy stan, zwiększając szansę osiągnięcia przy okazji innych podcelów, w danej chwili nie rozpatrywanych.

6. Funkcja

$$g_6(l_{Co}, l_o) = \begin{cases} 0 & , \text{ gdy } O_p = \emptyset \\ \frac{l_o - l_{Co}}{l_o} & , \text{ gdy } O_p \neq \emptyset \end{cases}$$

gdzie: l_o - jak w g_5 , l_{Co} określa łączną ilość formuł w zbiorach Co reguł przekształcania w podzbiorze O_p . Funkcja różnicuje podcele (wierzchołki) pod względem ilości nowych podcelów, które powstaną w wyniku ewentualnej rozbudowy DP od danego wierzchołka. Bardziej celowe jest rozbudowywanie DP od wierzchołków, wprowadzających mniejsze rozgałęzienie DP, między innymi ze względu na dążenie do minimalizowania zajętości pamięci maszyny cyfrowej.

Funkcję oceny g określono jako liniową kombinację wymienionych funkcji składowych:

$$g = \sum_{j=1}^6 \lambda_j \cdot g_j,$$

gdzie λ_j określa współczynnik wagowy dla funkcji składowej g_j . W zbiorze argumentów funkcji g można wydzielić zmienne, które dla danego wierzchołka p w DP mają wartość stałą podczas dalszej rozbudowy DP, mogą one zatem być wyznaczone jednokrotnie. Zmienne te tworzą zbiór:

$$\{v_p, f_p, k_p, t_o, t_p, l_o, l_{Ad}, l_{Co}\}.$$

Wartości zmiennych m_p oraz m_c ulegają natomiast zmianie po kolejnych krokach analizy, stąd potrzeba wielokrotnego wyznaczania wartości tych zmiennych dla danego wierzchołka p w DP.

Współczynniki wagowe λ_j , $j=1,2,\dots,6$ decydują o udziale poszczególnych funkcji składowych w funkcji g . Ich wartości mogą być na stałe określone arbitralnie przez użytkownika, na podstawie doświadczeń, zebranych podczas jej stosowania. Mogą też być zmieniane automatycznie przez dodatkowy mechanizm adaptacyjny metody, w toku jej wykorzystywania do kolejnych zadań danej klasy.

5.3.2. Funkcja oceny dla wierzchołków w drzewie rozwiązania

Dla oceny każdego wierzchołka S_i (odpowiadającego pewnej sytuacji S_i) w drzewie rozwiązania DR, pod kątem celowości przeprowadzenia względem niego analizy DP, wprowadzono funkcję oceny h w dwóch alternatywnych postaciach: h_1 i h_2 .

1. Funkcja

$$h_1(c_{S_{i-1}}, c_{S_i}^{O_1}) = c_{S_{i-1}} + c_{S_i}^{O_1},$$

gdzie: $C_{S_{i-1}}$ jest wartością funkcji h_1 dla wierzchołka S_{i-1} , poprzedzającego w DR dany wierzchołek S_i (dla wierzchołka początkowego S_p w DR funkcja $h_1 = C_{S_p} = 0$), $c_{S_i}^{O_1}$ określa koszt użycia reguły przekształcania O_1 do utworzenia sytuacji S_i z sytuacji S_{i-1} . Funkcja ta wyraża łączny koszt przejścia od wierzchołka początkowego S_p do danego wierzchołka S_i , po istniejącej ścieżce w DR, czyli łączny koszt sekwencji przekształcającej od sytuacji S_p do sytuacji S_i , przy użyciu wskazanych w DR reguł przekształcania. Wartość funkcji h_1 , wyznaczona dla danego wierzchołka S_i , nie ulega zmianie podczas dalszych kroków metody.

2. Funkcja

$$h_2(S_i, S_k) = \frac{|S_k \setminus S_i|}{|S_k|},$$

gdzie: S_i jest zbiorem formuł, opisujących sytuację S_i , S_k jest zbiorem formuł, opisujących cel S_k . Funkcja ta pozwala określić stopień zbliżenia sytuacji S_i do celu S_k . Jeśli w S_i brak każdej z formuł, występujących w opisie celu S_k , to $h_2=1$. Jeśli S_i zawiera wszystkie formuły z S_k , to $h_2=0$. W ogólnym przypadku $h_2 \rightarrow 0$, gdy S_i zmierza do zrównoważenia wszystkich swych formuł z formułami celu S_k . Wartość funkcji h_2 , wyznaczona dla danego wierzchołka S_i , nie ulega zmianie podczas dalszych kroków metody.

Użycie funkcji h_1 jako funkcji oceny dla wierzchołków w DR jest wskazane wówczas, gdy poszukiwana jest sekwencja przekształcająca o najmniejszym łącznym koszcie. Wymagana jest wówczas znajomość kosztu poszczególnych przekształceń w zależności od wartości argumentów reguł przekształcania.

Użycie funkcji h_2 prowadzi do poszukiwania sekwencji przekształcającej poprzez wykonanie możliwie małej ilości kroków metody. Zalecane jest wówczas, gdy nieistotny jest koszt

przekształcenia, natomiast ważna jest szybkość uzyskania rozwiązania.

5.3.3. Opis kroków wariantu C metody dwukierunkowego poszukiwania sekwencji przekształcającej w przestrzeni stanów, z użyciem strategii uporządkowanego wyboru

W przedstawionym poniżej wariantcie C metody drzewo podcełów rozbudowywane jest w kierunku, wyznaczonym maksymalną wartością funkcji oceny g . O kierunku rozwoju drzewa rozwiązania decyduje minimalna wartość funkcji h .

① Sprawdzić, czy $S_k \subseteq S_p$. Jeśli tak - stwierdzono istnienie celu w sytuacji początkowej (błąd w sformułowaniu zadania) - przejść do ⑬.

② Utworzyć listę O kolejnych reguł przekształcania (kolejność reguł określona arbitralnie). Dla każdej reguły O_1 obliczyć:

- $l_{Ad}^{O_1}$ (ilość formuł Ad w opisie reguły),
- $l_{Co}^{O_1}$ (ilość formuł Co w opisie reguły),
- $l_o^{O_1}$ (łącznie ilość formuł w opisie reguły).

Założyć ograniczenia ilościowe (tzw. limity) dla metody, w postaci maksymalnych ilości:

- n_{max} wierzchołków w DP oraz m_{max} wierzchołków w DR,
- k_{max} przebiegów analizy DP (przebiegiem analizy jest ciąg akcji od rozpoczęcia analizy do jej niepowodzenia lub sukcesu),

które wystąpią podczas poszukiwania sekwencji przekształcającej $\mathcal{T}$.

③ Na podstawie opisu celu S_k utworzyć zbiór koniunkcyjny wierzchołków początkowych DP. Utworzyć listę W z tymi wierzchołkami. Dla każdego wierzchołka p_i z listy W wyz-

naczyć zbiór N_{p_i} numerów w liście 0 tych reguł przekształcania, za pomocą których można rozbudować DP od danego wierzchołka. Przyjąć dla każdego wierzchołka z listy W: $m_p=0$, $v_p=1$. Przyjąć $m_c=0$.

Utworzyć wierzchołek początkowy S_p w DR na podstawie opisu sytuacji początkowej S_p . Utworzyć listę S z tym wierzchołkiem. Obliczyć wartość funkcji h dla wierzchołka S_p .

- ④ Wyznaczyć dla każdego wierzchołka z listy W wartość funkcji g. Wybrać z listy W wierzchołek p, posiadający maksymalną wartość funkcji oceny g, usuwając go równocześnie z W. Dla wybranego wierzchołka p, na podstawie listy 0 oraz zbioru N_p utworzyć listę ON reguł przekształcania, za pomocą których można rozbudować DP od wierzchołka p. Jeśli lista ON jest pusta - przejść do ⑭.

- ⑤ Wybrać z listy ON pierwszą regułę O_1 , umożliwiającą rozbudowanie DP od wierzchołka p. Utworzyć puste listy WN i SN.

- ⑥ Jeśli w liście ON brak takiej reguły, przejść do ⑬.
Dla wierzchołka p oraz reguły O_1 utworzyć zbiór wektorów podstawień: $\{v_1^{*1}, v_1^{*2}, \dots, v_1^{*r}\}$, $r \geq 1$. Utworzyć listę VN z kolejnymi wektorami podstawień.

- ⑦ Wybrać z listy VN pierwszy wektor podstawienia v_1^* , usuwając go jednocześnie z VN. Rozbudować DP od wierzchołka p, za pomocą reguły O_1 i wektora v_1^* . Dopisać do listy WN zbiór wierzchołków DP, powstałych podczas rozbudowy DP w tym kroku.

Jeśli podczas rozbudowy DP osiągnięto limit $n_{\max}$ - przejść do ⑮.

- ⑧ Uporządkować elementy listy S wg rosnącej wartości funkcji oceny h. Wybrać wierzchołek S_1 , będący pierwszym elementem listy S.

- ⑨ Przeprowadzić analizę DP od pierwszego z ostatnio dopisanych do listy WN wierzchołków, względem wierzchołka S_i (p. 4.3). Dopisywać do listy SN wierzchołki DR, powstające w wyniku analizy DP, obliczając równocześnie wartość funkcji oceny h dla tych wierzchołków.
- Jeśli podczas budowy DR osiągnięto limit $m_{\max}$ - przejść do ⑩.
- Jeśli analiza DP doprowadziła do sukcesu, czyli do osiągnięcia celu głównego w DP - przejść do ⑪.
- Zliczyć niepowodzenie analizy DP ($m_c := m_c + 1$), zwiększyć o 1 wartości parametrów m_p dla wierzchołków, które spowodowały niepowodzenie analizy.
- ⑩ Jeśli osiągnięto limit $k_{\max}$ - przejść do ⑩.
- ⑪ Jeśli wierzchołek S_i nie był ostatnim elementem listy S , wybrać następny element listy S jako wierzchołek S_i i przejść do ⑨.
- ⑫ Dołączyć listę SN do końca listy S , usuwając jednocześnie z SN wszystkie jej elementy. Wybrać z listy ON kolejną regułę O_1 , umożliwiającą rozbudowanie DP od wierzchołka p . Przejść do ⑥.
- ⑬ Dla każdego wierzchołka p_i z listy WN wyznaczyć zbiór N_{p_i} numerów w liście O tych reguł przekształcania, za pomocą których można rozbudować DP od danego wierzchołka.
- Dołączyć listę WN do końca listy W .
- ⑭ Jeśli lista W nie jest pusta - przejść do ④.
- ⑮ Sekwencja przekształcająca $\mathcal{T}$ nie istnieje. Przejść do ⑮.
- ⑯ Nie znaleziono sekwencji przekształcającej $\mathcal{T}$ dla zadanych limitów. Przejść do ⑮.
- ⑰ Odczytać sekwencję przekształcającą $\mathcal{T}$ z DR jako ścieżkę od wierzchołka początkowego S_p do wierzchołka, utworzonego w wyniku osiągnięcia celu głównego w DP.
- ⑱ Koniec działania metody.

Przedstawiony wariant C metody w ogólnym przypadku umożliwia znalezienie sekwencji przekształcającej $\mathcal{T}$ w sposób bardziej efektywny niż warianty A i B. Rozwój DP w kierunku, wyznaczonym maksymalną wartością funkcji oceny g , jest wyrazem dążenia do szybkiego wygenerowania niezbędnego fragmentu DP, umożliwiającego sukces analizy DP. Wiąże się to ściśle z chęcią istotnego ograniczenia rozrostu DP, czyli ograniczenia zajętości pamięci maszyny cyfrowej. Rozwój DR w kierunku, wyznaczonym minimalną wartością funkcji oceny h jest wyrazem dążenia do uzyskania rozwiązania o jak najmniejszym koszcie (w przypadku użycia funkcji h_1) lub do uzyskania rozwiązania w jak najkrótszym czasie (w przypadku użycia funkcji h_2).

Efekt działania funkcji h_1 można wykorzystać przy użyciu metody do generowania ciągu rozwiązań alternatywnych, np. w celu wybrania rozwiązania o najmniejszym koszcie. Wówczas, po utworzeniu pierwszego rozwiązania $\mathcal{T}_1$, można z listy S usunąć wszystkie wierzchołki w DR, którym odpowiada koszt większy niż koszt rozwiązania $\mathcal{T}_1$. Odpowiada to odcinaniu w DR całych wiązek gałęzi nie prowadzących do rozwiązań lepszych - z punktu widzenia kosztu - niż $\mathcal{T}_1$. Odcinanie dalszych wiązek w DR następuje każdorazowo po znalezieniu kolejnego rozwiązania o koszcie mniejszym niż koszt rozwiązania, uznawanego dotychczas za najlepsze.

Takie postępowanie zapewnia zbieżność do rozwiązania quasi-optimalnego w procesie poszukiwania sekwencji przekształcającej. Zapewnia również wymienność dwóch ważnych parametrów procesu poszukiwania rozwiązania: jakości rozwiązania oraz czasu uzyskania rozwiązania.

5.4. Automatyczna modyfikacja funkcji oceny jako mechanizm adaptacyjny metody

Z myślą o wielokrotnym stosowaniu wariantu C metody do poszukiwania rozwiązań różnych zadań z pewnej klasy zadań, wprowadzono do metody mechanizm adaptacyjny, polegający na automatycznej modyfikacji funkcji oceny g . Ma on głównie na celu sukcesywne zwiększanie szybkości uzyskiwania rozwiązań zadań podobnych do wcześniej rozwiązywanych. Działa w oparciu o informacje o znalezionych rozwiązaniach, zgromadzone i uogólnione do postaci, umożliwiającej wykorzystanie ich jako doświadczeń w działaniach przyszłych.

Algorytm automatycznej modyfikacji funkcji oceny g zostanie przedstawiony w postaci sekwencji działań, które wykonywane są każdorazowo po znalezieniu rozwiązania. Efektem tych działań będzie zmodyfikowanie wartości współczynników λ_k , $k=1,2,\dots,6$, w funkcji g . Wcześniej jednak, podczas działania metody, niezbędne jest tworzenie na bieżąco, w kroku 4. metody, zbioru Z kolejnych list W wierzchołków, od których można rozbudowywać DP w danym kroku metody, wraz z wartościami funkcji g oraz wartościami funkcji składowych $g_1, g_2, \dots, g_6$ dla każdego wierzchołka z listy W . Algorytm wykorzystuje też, zbudowane podczas działania metody, drzewa: DP i DR oraz użyte w tym działaniu wartości współczynników λ_k , $k=1,2,\dots,6$.

- ① Na podstawie znalezionej sekwencji przekształcającej $\mathcal{T}$:

$$S_0 \longrightarrow S_1 \longrightarrow S_2 \longrightarrow \dots \longrightarrow S_i \longrightarrow S_{i+1} \longrightarrow \dots \longrightarrow S_m,$$

$$S_0 = S_p, S_k \subseteq S_m,$$

utworzyć listę WW wierzchołków w DP,

$$WW = \{p_1^*, p_2^*, \dots, p_i^*, \dots, p_m^*\}, p_i^* \longleftrightarrow S_{m-i+1}, i=1,2,\dots,m.$$

- ② Na podstawie zbioru Z oraz listy WW utworzyć zbiór Z^* ,

$$Z^* = \{W^1, W^2, \dots, W^j, \dots, W^m\}, Z^* \subset Z,$$

gdzie W^j jest pierwszą z list w zbiorze Z , w której występuje wierzchołek $p_j^* \in WW$ ($j=1, 2, \dots, m$).

- ③ Dla każdej listy $W^j \in Z^*$, $j=1, 2, \dots, m$, obliczyć oceny $oc^j(g_k)$ skuteczności składowych g_k , $k=1, 2, \dots, 6$, funkcji g , względem wierzchołka p_j^* . W celu obliczenia $oc^j(g_k)$ należy:

- a) uporządkować listę $W^j = \{p_0, p_1, \dots, p_{n_j}\}$ względem rosnących wartości składowej g_k funkcji g - powstaje lista W_u^j ,
 b) określić numer l_j wierzchołka p_j^* w liście W_u^j ($0 \leq l_j \leq n_j$),
 c) obliczyć

$$oc^j(g_k) = \frac{l_j}{n_j}.$$

Oceny $oc^j(g_k)$, $k=1, 2, \dots, 6$, $j=1, 2, \dots, m$, są liczbami z przedziału $[0, 1]$. Ocena $oc^j(g_k)=1$, gdy w liście W^j maksimum składowej g_k występuje dla wierzchołka p_j^* ,
 $oc^j(g_k)=0$, gdy w liście W^j wierzchołkowi p_j^* odpowiada minimum składowej g_k funkcji g .

- ④ Na podstawie obliczonych ocen wyznaczyć średnie oceny $oc_{sr}(g_k)$:

$$oc_{sr}(g_k) = \frac{1}{m} \sum_{j=1}^m oc^j(g_k), k=1, 2, \dots, 6.$$

- ⑤ Na podstawie dotychczasowych wartości współczynników λ_k , $k=1, 2, \dots, 6$, oraz na podstawie obliczonych średnich ocen skuteczności funkcji składowych g_k , wyznaczyć nowe wartości współczynników λ_k :

$$r = \min_k \{\lambda_k \cdot oc_{sr}(g_k)\},$$

$$s = \max_k \{\lambda_k \cdot oc_{sr}(g_k)\},$$

n - ilość przeprowadzonych dotychczas modyfikacji współczynników λ_k względem danej klasy zadań,

$$\lambda_k := \frac{n \cdot \lambda_k + \frac{\lambda_k \cdot oc_{sr}(g_k) - r}{s - r}}{n + 1}, \quad k=1,2,\dots,6 \text{ - dla } s-r>0,$$

Dla $s-r=0$ wartości współczynników λ_k nie ulegają zmianie.

⑥ Koniec algorytmu.

Idea przedstawionego algorytmu modyfikacji funkcji oceny g wynika ze spostrzeżenia, że z idealną trafnością oceny wierzchołków w DP za pomocą funkcji g ("skutecznością funkcji g ") mielibyśmy do czynienia wówczas, gdyby każdorazowo w 4. kroku metody wybierany był wierzchołek p_j^* , właściwy z punktu widzenia przyszłego rozwiązania. Wówczas drzewo podcelów DP powstawałoby tylko w niezbędnych dla uzyskania rozwiązania rozmiarach. Z kolei na skuteczność funkcji g wpływają skuteczności funkcji składowych g_k , $k=1,2,\dots,6$. Miarą skuteczności danej składowej g_k w j -tym kroku jest ocena $oc^j(g_k)$, wynikająca - jak to przedstawiono w 3. kroku algorytmu modyfikacji - z porównania wartości funkcji g_k dla wierzchołka p_j z wartościami, obliczonymi dla innych wierzchołków w DP. Miarą globalnej skuteczności składowej g_k w procesie poszukiwania rozwiązania jest ocena średnia $oc_{sr}(g_k)$. Celowe jest zwiększenie roli w funkcji g tych funkcji składowych g_k , które uzyskują wyższe oceny średnie, tak, jak celowe jest zmniejszenie roli w funkcji g tych składowych g_k , które uzyskują niższe oceny średnie.

W zależności, służącej do wyznaczania wartości nowych współczynników λ_k w 5. kroku, uwzględniono znaczny wpływ poprzednich wartości λ_k , wynikających z dokonanych wcześniej ocen, zmniejszając wpływ oceny w danym przebiegu algorytmu. Ma to zapewnić większą stabilność tendencji zmian wartości współczynników λ_k w procesie modyfikacji i zmniejszenie wpływu takich przypadków, które występują rzadko i dążą do rady-

kalnych zmian wartości współczynników wagowych λ_k w funkcji oceny g .

5.5. Uwagi na temat możliwych zastosowań metody

Opracowana metoda, jakkolwiek ogólna, może jednak znaleźć zastosowanie w różnych, konkretnych dziedzinach, w których istnieje możliwość powierzenia komputerowi zadania automatycznego poszukiwania sekwencji przekształceń. Opracowany, prosty język opisu zadania dla metody umożliwia przedstawianie zadań, zawierających się w modelu, opisanym w rozdziale 3. Wśród problemów, dających się sformułować w sposób, umożliwiający zastosowanie metody, można wyróżnić problemy automatycznego planowania działań robotów ([13],[19],[46],[49]), problemy automatycznego i wspomaganego projektowania ([27],[28],[38],[53]) oraz automatycznego programowania procesów sekwencyjnych, np. procesów technologicznych, przedstawianych w postaci programów dla obrabiarek sterowanych numerycznie ([17],[19],[39],[60],[63]) - temat ten porusza również p. 6.6 niniejszej rozprawy.

Użyte w metodzie funkcje ocen, podobnie jak i sama metoda, nie są zorientowane na konkretne, specjalistyczne zastosowanie, lecz ilustrują możliwość wpływania na proces poszukiwania rozwiązania, poprzez sukcesywne doskonalenie strategii działania metody w wieloetapowym procesie uczenia się rozwiązywania zadań podobnych. Zaproponowany algorytm automatycznej modyfikacji funkcji oceny g może być użyty również w odniesieniu do innych niż użyta w metodzie funkcji ocen, wyrażających się liniową kombinacją pewnych funkcji składowych.

6. PROGRAMOWA REALIZACJA PROPONOWANEJ METODY

6.1. Informacje wstępne

Na oprogramowanie opisanej w rozprawie metody składa się:

- 1° zmodyfikowana wersja systemu wspomagania dydaktyki DIALOS/INT [5,34,35],
- 2° wykonane modularnie względem systemu DIALOS/INT dodatkowe procedury wraz z ich bazą danych.

Do wykorzystania systemu DIALOS/INT w oprogramowaniu przedstawionej tu metody upoważnia autora rozprawy współudział, obok mgra inż. Artura Klajna, w pracach projektowych, programowych, dokumentacyjnych oraz eksploatacyjnych podczas budowy systemu, w tym między innymi wyłączne autorstwo tej części systemu, która stanowi oprogramowanie metody poszukiwania rozwiązań zadania (ROZWIĄZYWACZ wraz z jego bazą danych). Stąd opis oprogramowania metody będzie się składać również z opisu najistotniejszych elementów systemu DIALOS/INT.

Całość oprogramowania będzie dalej nazywana systemem poszukiwania rozwiązań (SPR) lub krótko: systemem. Opis systemu zawiera:

- informacje o zasadach korzystania z systemu,
- informacje o strukturze programowej systemu,
- przykład użycia systemu do rozwiązywania zadań.

6.2. Przeznaczenie i ogólne zasady korzystania z systemu poszukiwania rozwiązań

System SPR przeznaczony jest do wspomagania rozwiązywania zadań, zdefiniowanych w p. 3.1 niniejszej rozprawy, nazywanych w systemie zadaniami problemowymi.

Rozwiązaniem zadania problemowego ma być sekwencja $\mathcal{T}$ przekształcająca określoną sytuację początkową w zadaną sytuację

docelową (p. 3.1). Wprowadzony do systemu opis zadania problemowego powinien zawierać: opis sytuacji początkowej, istotny dla problemu fragment opisu sytuacji docelowej oraz opis reguł przekształcania. Opisy te podawane są za pomocą języka opisu zadań (JOZ)(p. 3.2, p. 3.3) oraz komend organizacyjnych systemu. Elementy zadania problemowego mogą być aktualizowane przez użytkownika w czasie procesu poszukiwania rozwiązania (przy pracy w trybie interakcyjnym). Przebieg rozwiązywania zadania jest monitorowany na wybranym poziomie szczegółowości.

Do poszukiwania rozwiązania stosowana jest w systemie metoda poszukiwania sekwencji przekształcającej w przestrzeni stanów, opisana w niniejszej rozprawie, z dodatkowym rozwinięciem o możliwość interakcyjnej współpracy użytkownika z systemem.

Opis zadania problemowego i przebieg rozwiązywania (wraz z informacjami o wprowadzanych podczas interakcyjnego poszukiwania rozwiązania zmianach opisów elementów zadania) posiadają swoje reprezentacje w odpowiednich kartotekach systemu. W dowolnym momencie proces rozwiązywania może być przerwany na życzenie użytkownika i następnie kontynuowany w innym terminie. Następuje wówczas składowanie stanu rozwiązania i przechowanie go do chwili rekonstrukcji.

Dla zilustrowania metody oraz jej zastosowań wdrożono w systemie zadania typu manipulacyjnego. O wyborze tego typu zadań zadecydowały ich walory dydaktyczne. Dzięki prostej fizycznej interpretacji mogą one stanowić wprowadzenie do bardziej złożonych typów zadań. Ponadto mogą znaleźć zastosowanie między innymi w procesach projektowania i w robotyce.

6.3. Język komunikacji użytkownika z systemem

Komunikację użytkownika z systemem umożliwiają następujące języki:

- język komend (JK) służy użytkownikowi do organizacji dialogu z systemem (jest to więc język komunikacji organizacyjnej),
- komunikacja w zakresie merytorycznym realizowana jest za pomocą języka opisu zadań (JOZ),
- język komunikatów systemu.

6.3.1. Język komunikacji organizacyjnej

Podstawowym elementem języka komunikacji organizacyjnej jest komenda. Ogólna postać komendy jest następująca:

:komenda_lista parametrów

gdzie: "komenda" jest słowem kluczowym (z repertuaru 50 słów kluczowych), określającym rodzaj polecenia, "lista parametrów" stanowi zbiór wartości precyzujących dane polecenie.

Lista ta może nie wystąpić w pewnych komendach. Znak ":", poprzedzający słowo kluczowe, pozwala odróżnić komendę organizacyjną od zdania z informacją merytoryczną.

Każda z komend posiada określoną postać skróconą, która może być użyta w zastępstwie słowa kluczowego.

Wśród komend najważniejszą dla użytkownika grupę stanowią komendy, używane podczas definiowania zadania problemowego oraz podczas poszukiwania rozwiązania. Wykaz tych wybranych komend zamieszczono w tabeli 6.1.

Znaczenia poszczególnych komend określają funkcje realizowane przez poszczególne procedury systemu. Wśród komend występują między innymi komendy inicjujące i kończące zadania obliczeniowe (przetwarzanie) w określonych trybach pracy (pozycje 2 + 7), komendy dla wprowadzania opisu zadania problemowego

Tabela 6.1 Zestaw wybranych komend systemu poszukiwania rozwiązań SPR

Lp	Komenda	Skrót	Funkcja	Format	Kontekst dozwolony
1	2	3	4	5	6
1	:FINISH	:FI	Zakończenie pracy modułu REALIZATOR	:FINISH	NO USER
2	:INPUT	:IN	Inicjacja zadania obliczeniowego - wprowadzanie informacji do systemu SPR	:INPUT DIAL-INT-yyy gdzie: yyy jest numerem użytkownika w systemie	NO USER
3	:JOB	:JB	Inicjacja zadania obliczeniowego - wprowadzanie informacji do systemu i rozwiązywanie zadań problemowych w trybie automatycznym, wsadowo	:JOB DIAL-INT-yyy	NO USER
4	:LOGIN	:LN	Inicjacja zadania obliczeniowego - wprowadzanie informacji do systemu i rozwiązywanie zadań problemowych w trybie interakcyjnym, konwersacyjnie	:LOGIN DIAL-INT-yyy	NO USER
5	:ENDINPUT	:EI	Zakończenie zadania obliczeniowego, zainicjowanego komendą :INPUT	:ENDINPUT	INPUT

Tabela 6.1 - c.d.

1	2	3	4	5	6
6	:ENDJOB	:EJ	Zakończenie zadania obliczeniowego, zainicjowanego komendą :JOB	:ENDJOB	JOB
7	:LOGOUT	:LT	Zakończenie zadania obliczeniowego, zainicjowanego komendą :LOGIN	:LOGOUT	LOGIN
8	:FORMULA	:FM	Inicjacja wprowadzania treści formuły	:FORMULA treść gdzie: treść - jest treścią formuły	INPUT, JOB, LOGIN, INITIAL, GOAL, OPERATOR, ENVIRONMENT, SOLVING, CO-OPERATOR, CA-OPERATOR, AD-OPERATOR, RE-OPERATOR
9	:INITIAL	:IT	Inicjacja wprowadzania opisu sytuacji początkowej zadania problemowego lub dołączenie sytuacji początkowej do definiowanego zadania	:INITIAL a gdzie: a jest nazwą sytuacji początkowej lub :INITIAL *a gdzie: a jest nazwą sytuacji początkowej lub środowiska problemowego	INPUT, JOB, LOGIN, EXERCISE

Tabela 6.1 - c.d.

1	2	3	4	5	6
10	:GOAL	:GL	Inicjacja wprowadzania opisu sytuacji docelowej zadania problemowego lub dołączenie sytuacji docelowej do definiowanego zadania	:GOAL a lub :GOAL *a gdzie: a jest nazwą sytuacji docelowej	INPUT, JOB, LOGIN, EXERCISE
11	:ENVIRON	:EV	Inicjacja wprowadzania opisu środowiska problemowego zadania lub grupy zadań	:ENVIRON a lub :ENVIRON *a gdzie: a jest nazwą środowiska	INPUT, JOB, LOGIN, EXERCISE
12	:OPERATOR	:OP	Inicjacja wprowadzania opisu reguły przekształcania lub dołączenie reguły przekształcania do opisu zadania	:OPERATOR a lub :OPERATOR *a ₁ , ..., *a _k gdzie: a oraz a _i , i=1,2,...,k jest nazwą operatora	INPUT, JOB, LOGIN, EXERCISE, SOLVING
13	:COND	:CO	Inicjacja wprowadzania opisu zbioru Co reguły przekształcania	:COND	OPERATOR
14	:CANC	:CA	Inicjacja wprowadzania opisu zbioru Ca reguły przekształcania	:CANC	OPERATOR

Tabela 6.1 - c.d.

1	2	3	4	5	6
15	:ADD	:AD	Inicjacja wprowadzania opisu zbioru Ad reguły przekształcania	:ADD	OPERATOR
16	:REL	:RE	Inicjacja wprowadzania opisu zbioru Re reguły przekształcania	:REL	OPERATOR
17	:EXERCISE	:EX	Inicjacja wprowadzania opisu zadania problemowego	:EXERCISE a	INPUT, JOB, LOGIN
18	:PREDICAT	:PD	Deklaracja określonych nazw jako nazw predykatów	:PREDICAT $a_1, \dots, a_k$ gdzie: $a_i, i=1, 2, \dots, k$ jest nazwą predykatu	INPUT, JOB, LOGIN, FORMULA, INITIAL, GOAL, ENVIRONMENT, OPERATOR, EXERCISE, CO-OPERATOR, CA-OPERATOR, AD-OPERATOR, RE-OPERATOR
19	:CCONSTANT	:CN	Deklaracja określonych nazw jako nazw stałych	:CONSTANT $a_1, \dots, a_k$ gdzie: $a_i, i=1, 2, \dots, k$ jest nazwą stałej	jak dla komendy :PREDICAT
20	:VARIABLE	:VA	Deklaracja określonych nazw jako nazw zmiennych	:VARIABLE $a_1, \dots, a_k$ gdzie: $a_i, i=1, 2, \dots, k$ jest nazwą zmiennej	jak dla komendy :PREDICAT

Tabela 6.1 - c.d.

1	2	3	4	5	6
21	:END	:EN	Zakończenie wprowadzania opisu określonego fragmentu zadania problemowego	:END xx gdzie: xx jest jednym z symboli: IT, GL, EV, OP, CO, CA, AD, RE, EX	jeden z kontekstów: INITIAL, GOAL, ENVIRONMENT, OPERATOR, EXERCISE, CO-OPERATOR, CA-OPERATOR, AD-OPERATOR, RE-OPERATOR - w zależności od xx
22	:BREAK	:BR	Wstrzymanie aktualnie wykonywanej czynności	:BREAK	FORMULA, INITIAL, GOAL, ENVIRONMENT, OPERATOR, EXERCISE, CO-OPERATOR, CA-OPERATOR, AD-OPERATOR, RE-OPERATOR, SOLVING
23	:QUIT	:QU	Przerwanie czynności wstrzymanej komendą :BREAK	:QUIT	BREAK IN
24	:CONTINUE	:CU	Kontynuacja czynności wstrzymanej komendą :BREAK	:CONTINUE	BREAK IN
25	:PRINT	:PR	Wyprowadzanie treści formuły, wyprowadzanie treści wszystkich formuł w kartotece AFILE lub wyprowadzanie fragmentu opisu zadania problemowego	:PRINT *nr gdzie: nr jest numerem formuły, lub :PRINT *ALL /gdy wszystkie formuły/, lub	INPUT, JOB, LOGIN, BREAK IN, SOLVING

Tabela 6.1 - c.d.

1	2	3	4	5	6
				:PRINT a gdzie: a jest nazwą fragmentu opisu zadania problemowego	
26	:TYPE	:TP	Wyprowadzanie typu nazwy uży- tej w systemie	:TYPE a gdzie: a jest nazwą	wszystkie, za wyjąt- kiem NO USER
27	:CONTEXT	:CX	Wyprowadzanie informacji o aktualnym kontekście	:CONTEXT	wszystkie
28	:DO	:DO	Inicjowanie rozwiązywania zadania problemowego	:DO a gdzie: a jest nazwą zadania	JOB, LOGIN
29	:RETURN	:RT	Określenie ilości akcji sys- temu, po wykonaniu których sterowanie przekazywane jest użytkownikowi	:RETURN k gdzie: k jest liczbą całkowitą	JOB, LOGIN, SOLVING
30	:LIMIT	:LM	Określenie maksymalnej ilość- ci akcji systemu podczas rozwiązywania zadania	:LIMIT k gdzie: k jest liczbą całkowitą	JOB, LOGIN, SOLVING
31	:REPORT	:RP	Określenie poziomu protokoło- wania	:REPORT k gdzie: k=0,1,2	JOB, LOGIN, SOLVING
32	:GO	:GO	Uruchomienie procedury poszu- kiwania rozwiązania	:GO	SOLVING

Tabela 6.1 - c.d.

1	2	3	4	5	6
33	:WRITE	:WR	Wyprowadzanie opisu sytuacji bieżącej, podcelu lub formuły w opisie sytuacji	:WRITE xk gdzie: x jest jednym z symboli: S/sytuacja bieżąca/, P/podcel/, E/formuła/; k jest numerem konstrukcji, określonej przez x	LOGIN, SOLVING
34	:SEQUENCE	:SE	Wyprowadzanie opisu ciągu przekształceń, prowadzących od sytuacji początkowej do wskazanej sytuacji	:SEQUENCE Sk gdzie: k jest numerem sytuacji	LOGIN, SOLVING
35	:ANNEX	:AN	Dołączenie reguły przekształcania do zadania lub dołączenie formuły do opisu sytuacji	:ANNEX a gdzie: a jest nazwą reguły przekształcania lub :ANNEX *k,Sm gdzie: k jest numerem formuły, m jest numerem sytuacji	SOLVING
36	:EXCLUDE	:EC	Wyłączenie reguły przekształcania z zadania lub wyłączenie formuły z opisu sytuacji	:EXCLUDE a gdzie: a jest nazwą reguły przekształcania lub	SOLVING

Tabela 6.1 - c.d.

1	2	3	4	5	6
				:EXCLUDE *k,Sm gdzie: k jest numerem formuły, m jest numerem sytuacji	
37	:FORWARD	:FW	Przekształcenie wskazanej sytuacji za pomocą danej reguły przekształcania	:FORWARD a,Sk gdzie: a jest nazwą reguły przekształcania, k jest numerem sytuacji	SOLVING
38	:BACKWARD	:BW	Rozbicie wskazanego celu na podcele składowe, za pomocą danej reguły przekształcania	:BACKWARD a,Pk gdzie: a jest nazwą reguły przekształcania, k jest numerem celu	SOLVING
39	:SAVE	:SV	Składowanie stanu aktualnie rozwiązywanego zadania problemowego, zakończenie rozwiązywania	:SAVE	SOLVING
40	:RESTORE	:RS	Odzyskanie informacji o zadaniu, składowanym wcześniej komendą :SAVE, z zamiarem kontynuowania przerwanej rozwiązywania zadania	:RESTORE	JOB, LOGIN
41	:STRATEGY	:SY	Wybieranie strategii poszuki-	:STRATEGY s	JOB, LOGIN, SOLVING

Tabela 6.1 - c.d.

1	2	3	4	5	6
			wania rozwiązania w trybie automatycznym /wybór wariantu A, B lub C metody/	gdzie: s jest jednym z symboli wariantów metody: A, B lub C	JOB, LOGIN, SOLVING
42	:ALTER	:AL	Określenie nowych wartości współczynników wagowych λ_k , funkcji oceny g	:ALTER $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6$ gdzie: $\lambda_i, i=1,2,\dots,6$ jest liczbą rzeczywistą	JOB, LOGIN, SOLVING
43	:FUNCTION	:FN	Wyprowadzanie postaci funkcji oceny g - wyprowadzanie wartości współczynników wagowych - lub postaci funkcji oceny h /h=h ₂ /	:FUNCTION f gdzie: f jest jednym z symboli: G lub H	JOB, LOGIN, SOLVING
44	:MODIFY	:MO	Uruchomienie procedury automatycznej modyfikacji funkcji oceny g w module MCDYFIKATOR	:MODIFY *a,Sm gdzie: a jest nazwą zadania problemowego, m jest numerem sytuacji, w której osiągnięto analizowane rozwiązanie zadania a	JOB, LOGIN

Tabela 6.2. Wykaz kontekstów komend języka komunikacji
użytkownika z modułem REALIZATOR

Lp	Kontekst	Znaczenie
1	NO USER	Poza zadaniem systemu SPR
2	INPUT	W czasie wprowadzania opisu zadania problemowego lub jego części
3	JOB	W czasie przetwarzania w trybie partio- wym
4	LOGIN	W czasie przetwarzania w trybie konwer- sacyjnym
5	FORMULA	W czasie wprowadzania treści formuły
6	INITIAL	W czasie wprowadzania opisu sytuacji początkowej w zadaniu problemowym
7	GOAL	W czasie wprowadzania opisu fragmentu sy- tuacji docelowej w zadaniu problemowym
8	ENVIRONMENT	W czasie wprowadzania opisu środowiska problemowego zadania (grupy zadań)
9	OPERATOR	W czasie wprowadzania opisu reguły prze- kształcania w zadaniu problemowym
10	CO-OPERATOR	W czasie wprowadzania opisu warunków użycia reguły przekształcania
11	CA-OPERATOR	W czasie wprowadzania opisu wyrażeń wykreślanych reguły przekształcania
12	AD-OPERATOR	W czasie wprowadzania opisu wyrażeń dołączanych reguły przekształcania
13	RE-OPERATOR	W czasie wprowadzania opisu związków między argumentami reguły przekształcania
14	EXERCISE	W czasie wprowadzania opisu zadania problemowego do systemu
15	BREAK-IN	W czasie ingerencji
16	SOLVING	W czasie rozwiązywania zadania problemo- wego

(poz.: 8 + 21), komendy dla wyprowadzania informacji (poz.: 25 + 27, 33, 34, 43), komendy dla rozwiązywania problemów (poz.: 28 + 32, 35 + 38, 41, 42) oraz komenda dla modułu modyfikacji funkcji oceny (poz.: 44).

Po wprowadzeniu komendy procesor języka komend dokonuje jej identyfikacji a następnie odpowiednia procedura systemu wykonuje zawarte w niej polecenie [32]. Komendy mogą być wydawane systemowi w określonych warunkach zwanych kontekstami. Wykaz wszystkich kontekstów zamieszczono w tabeli 6.2.

6.3.2. Język komunikacji merytorycznej

Przy wyborze języka komunikacji merytorycznej kierowano się następującymi wymogami:

1. Język musi umożliwiać opisywanie dostatecznie szerokiej klasy zagadnień.
2. Posługiwanie się językiem nie może stwarzać użytkownikowi trudności.
3. Implementacja programowa języka musi być stosunkowo łatwa.

Wymogi te w zadowalającym stopniu spełnia język rachunku predykatów; na nim więc oparto język opisu zadań (p. 3.2), pełniący w systemie funkcję języka komunikacji merytorycznej.

Poszczególne elementy zadania problemowego przedstawiane są w postaci formuł (lub zbiorów formuł) tego języka. Formuły te, wprowadzone przez użytkownika do systemu, poddawane są analizie leksykalnej oraz syntaktycznej i w przypadku ich poprawności są zapamiętywane w bazie danych systemu. Na błędy w formułach system reaguje odpowiednimi komunikatami.

Wprowadzanie formuł JOZ może odbywać się w trybie konwersacyjnym bądź w trybie przetwarzania partiowego. Zaakceptowana przez system formuła otrzymuje identyfikujący ją numer (symbol), w celu ułatwienia użytkownikowi odwoływania się do

niej przy tworzeniu opisów złożonych konstrukcji. Reprezentacje wewnętrzne formuł mogą być przechowywane w systemie dowolnie długo, podobnie jak bazujące na nich fragmenty zadań lub całe zadania. Raz wprowadzona do systemu formuła może być użyta w wielu różnych konstrukcjach złożonych.

Formuły JOZ można wprowadzać w dowolnym momencie współpracy z systemem. W szczególności istnieje możliwość uzupełniania opisów poszczególnych konstrukcji w zadaniu o nowe elementy podczas rozwiązywania zadania, w celu przeprowadzenia redefinicji zadania.

W związku z ograniczonym zestawem znaków w maszynie cyfrowej wprowadzono następującą symbolikę:

-. negacja	+. alternatywa	&. koniunkcja
>. implikacja	=. równoważność	A. kwantyfikatory ogólny
E. kwantyfikatory szczegółowy		

Dla oznaczenia początku i końca zasięgu kwantyfikatora wprowadzono symbole: ' oraz ".

Zmienne, stałe i predykaty oznaczane są za pomocą nazw. Nazwą jest ciąg liter i ewentualnie cyfr, zaczynający się od litery, nie dłuższy niż 8 znaków. Nazwę wiąże się z typem elementu języka poprzez użycie odpowiedniej deklaracji (komendy: 18 + 20 w tabeli 6.1). Ilość argumentów predykatu jest określona przez pierwsze wystąpienie tego predykatu w formule.

Typy pewnych nazw są wstępnie określone. Nazwy C01, C02, ..., C99 oznaczają stałe, X01, X02, ..., X99 - zmienne, natomiast nazwy P01, P02, ..., P99 oznaczają predykaty. Ponadto nazwy Q1, Q2, ..., Q99 są zastrzeżone dla argumentów reguł przekształcania w zadaniu problemowym. Argumenty te występują w formułach ze zbiorów Co, Ca, Ad, Re reguły przekształcania w dowolnej kolejności i mają znaczenie lokalne w ramach opisu danej reguły. Argumenty mogą wystąpić w formule w miejsce

zmiennych.

Stałe liczbowe mogą mieć postać liczby całkowitej lub liczby rzeczywistej z kropką dziesiętną. Liczba (z ewentualnym znakiem) musi być poprzedzona znakiem # (np. #7, #-12.75).

Dla umożliwienia opisywania związków między argumentami reguł przekształcania (zbiór $Re \in O_1$) wprowadzono w systemie zbiór predykatów standardowych.

1. Dla przedstawiania i badania relacji arytmetycznych między argumentami liczbowymi reguły przekształcania zdefiniowano dwuargumentowe predykaty, których argumenty muszą zawierać stałe liczbowe:

ROW(x,y) - $x = y$

NRO(x,y) - $x \neq y$

MNI(x,y) - $x < y$

MNR(x,y) - $x \leq y$

WIR(x,y) - $x \geq y$

WIE(x,y) - $x > y$

2. Dla wyrażania funkcji dyskretnych wiążących argumenty reguły przekształcania zdefiniowano czteroargumentowy predykat

TAB(q_i, w_i, q_j, w_j),

w którym q_i, q_j są argumentami reguły przekształcania a w_i, w_j są wartościami argumentów q_i oraz q_j . Sens predykatu TAB można wyrazić następująco: "Dla $q_i=w_i, q_j=w_j$ oraz dla $q_j=w_j, q_i=w_i$ ".

Przykłady użycia JOZ zamieszczono w p. 6.6.

6.4. Organizacja procesu poszukiwania rozwiązania zadania problemowego

Poszukiwanie rozwiązania zadania w systemie SPR dokonywane jest w oparciu o ROZWIĄZYWACZ - zespół procedur, realizujący

metodę automatycznego poszukiwania sekwencji przekształcającej w przestrzeni stanów (rozdziały 4. i 5.).

System może automatycznie poszukiwać rozwiązania zadania, lub we współpracy z użytkownikiem, w trybie interakcyjnym. Automatyczne poszukiwanie rozwiązania przebiega według jednego z wariantów: A, B lub C metody. Drzewo podcelów DP oraz drzewo rozwiązania DR, powstające podczas rozwiązywania zadania, zapisywane są w kartotekach systemu i mogą być tworzone w sposób:

- automatyczny, tzn. sterowany przyjętą strategią, wg jednego z wariantów: A, B lub C,
- sterowany wyłącznie przez użytkownika,
- mieszany (częściowo automatyczny, częściowo sterowany przez użytkownika).

Szybkość znalezienia rozwiązania uzależniona jest przy automatycznym rozwoju przestrzeni poszukiwania rozwiązania od sposobu sformułowania zadania. W szczególności istotne znaczenie ma tu umiejętność dobrania opisów stanów obiektów w sytuacji początkowej i opisu celu (z pominięciem opisów nieistotnych dla rozwiązania), trafny wybór reguł przekształcania i ich opisów. Nadmiar informacji w opisie zadania problemowego może spowodować zbyt dynamiczny rozwój przestrzeni poszukiwania rozwiązania i - w konsekwencji - wydłużenie czasu oczekiwania na rozwiązanie lub wręcz osiągnięcie założonych limitów $n_{\max}$ i $m_{\max}$ (p. 5.1).

Użytkownik poszukujący rozwiązania zadania, śledząc rozwój przestrzeni poszukiwania rozwiązania, ma możliwość redefiniowania opisów poszczególnych elementów zadania podczas przebiegu procesu rozwiązywania. W szczególności ma on możliwość dołączenia opisów stanów do wskazanej sytuacji lub wyłączenia pewnych stanów, zdefiniowania brakującej reguły przekształca-

nia i dołączenia jej w dowolnym stanie procesu lub wyłączenia reguły przekształcania z zadania, itd. Dzięki temu użytkownik może wpływać na rozwój przestrzeni i przyspieszać uzyskanie rozwiązania. Użytkownik może również zmieniać wartości współczynników wagowych w funkcji oceny g (p. 5.3) podczas rozwiązywania zadania.

Rzeczywisty rozwój przestrzeni poszukiwania rozwiązania sterowany przez użytkownika następuje przez stosowanie wybranych reguł przekształcania do wskazanych sytuacji lub przez stosowanie reguł przekształcania do wskazanych podcelów. W rozwoju przestrzeni nie uczestniczy wówczas strategia systemu. System nie dopuszcza jednak do powtarzania czynności zawierających kroki, które zostały już wykonane i nie doprowadziły do rozwiązania. Informuje ponadto użytkownika, że trafił na rozwiązanie.

Rzeczywisty rozwój przestrzeni poszukiwania rozwiązania w sposób częściowo automatyczny, częściowo sterowany przez użytkownika, następuje w wyniku "złożenia" dwóch poprzednio przedstawionych trybów współpracy z systemem, z dowolną, indywidualizowaną w stosunku do potrzeb i różną w poszczególnych etapach procesu, przewagą każdej z nich. Możliwość ukierunkowywania rozwoju przestrzeni na podstawie obserwacji procesu, z możliwością ingerencji w jego przebieg, najpełniej wyraża zasadę interakcyjnej współpracy użytkownika z systemem.

Po znalezieniu przez system rozwiązania $\mathcal{T}$ pewnego zadania w trybie automatycznym (wg wariantu C metody) można zainicjować proces automatycznej modyfikacji funkcji oceny (p. 5.4), prowadzący do zmiany wartości współczynników λ_i , $i=1,2,\dots,6$ w funkcji g . Standardowe wartości współczynników λ_i wynoszą 1. Przystępując do rozwiązywania zadania wg wariantu C metody można każdorazowo wprowadzić nowe wartości współczynników λ_i ,

korzystać z wartości zmodyfikowanych (względem poprzednich wartości) przez procedurę modyfikacji lub korzystać z wartości standardowych.

6.5. Organizacja programowa systemu SPR

System SPR składa się z czterech modułów programowych oraz pewnej liczby kartotek systemowych. Strukturę systemu i jego związek z systemem DIALOS/INT przedstawiono na rysunku 6.1.

W skład systemu wchodzi następujące moduły:

1. Moduł inicjatora systemu (INICJATOR).
2. Moduł administratora systemu (ADMINISTRATOR).
3. Moduł realizatora systemu (REALIZATOR).
4. Moduł modyfikacji funkcji oceny (MODYFIKATOR).

Moduł INICJATOR służy do inicjacji systemu w danej instalacji obliczeniowej. Moduł ADMINISTRATOR służy do administrowania zbiorami danych, tworzącymi poszczególne kartoteki systemu oraz do administrowania użytkowaniem systemu. Szczegółowe informacje o funkcjach i sposobie użycia tych modułów zamieszczono w [34,35].

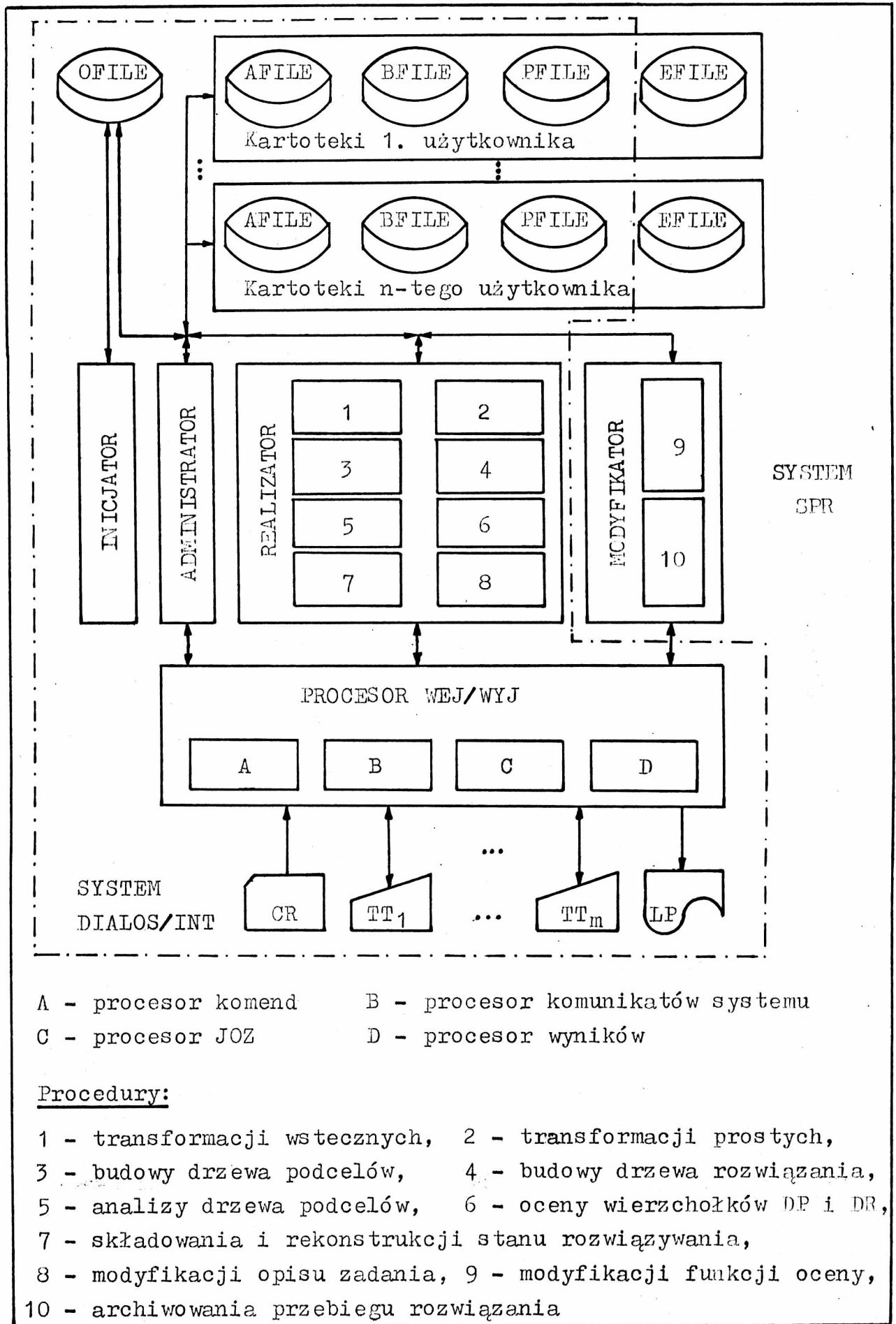
Funkcją modułu REALIZATOR jest wprowadzanie i rozwiązywanie zadań problemowych. Jego główną część stanowi ROZWIĄZYWACZ.

Moduł MODYFIKATOR służy do gromadzenia informacji o przebiegu rozwiązań zadań oraz do automatycznej modyfikacji funkcji oceny w metodzie poszukiwania sekwencji przekształcającej.

Bazę danych systemu SPR tworzą:

1. Kartoteka organizacyjna (OFILE).
2. Kartoteka analizy językowej i opisów zadań (AFILE).
3. Kartoteka rozwinięć wstecznych (BFILE).
4. Kartoteka rozwinięć progresywnych (PFILE).
5. Kartoteka doświadczeń systemu (EFILE).

Kartoteka organizacyjna jest jedna dla wszystkich użytkow-



Rys 6.1. Struktura systemu poszukiwania rozwiązań SPR i jego związek z systemem DIALOS/INT

ników systemu. Wszystkie pozostałe kartoteki są tworzone dla każdego użytkownika oddzielnie. Każda z kartotek 2 + 5 jest tworzona podczas wprowadzania użytkownika do systemu i może mieć jedną z dwóch możliwych wersji, różniących się wielkością zbioru dyskowego i wielkością poszczególnych obszarów kartotek. Kartoteki mają strukturę indeksową.

System SPR może być eksploatowany w trybie konwersacyjnym bądź partiowym, w konfiguracjach systemowych maszyn serii ODRA 1300, pod kontrolą systemu operacyjnego GEORGE 3. Korzystanie z systemu SPR ułatwiają makrokomendy: SPR oraz SPR-W (wraz z ich parametrami), zwalniając użytkownika od obowiązku znajomości szczegółów współpracy z systemem GEORGE 3. Makrokomenda SPR służy do obsługi przetwarzania systemu SPR w trybie konwersacyjnym, makrokomenda SPR-W - do obsługi przetwarzania w trybie partiowym. Między innymi obie makrokomendy obsługują składowanie kartotek systemu SPR na taśmy magnetyczne po zakończeniu przebiegu programu oraz obsługują rekonstrukcję potrzebnych kartotek systemu w pamięci dyskowej, na początku przetwarzania.

Postać źródłowa systemu SPR składa się z około 8000 instrukcji języka programowania FORTRAN 1900 i przechowywana jest, wraz z treścią źródłową makrokomend oraz binarną postacią oprogramowania systemu, na taśmie magnetycznej.

6.6. Przykład użycia systemu SPR do rozwiązywania zadania

Dla zilustrowania działania systemu sformułowano zadanie, które polega na wyznaczeniu ciągu przekształceń technologicznych (terminem "przekształcenie technologiczne" będzie dalej nazywana operacja technologiczna, bądź dowolna z jej składowych, takich jak: zabieg, czynność, ruch elementarny) przedsta-

wionego w uproszczeniu procesu technologicznego, realizowanego przy użyciu obrabiarki sterowanej numerycznie (OSN), czyli na znalezieniu "programu" dla sterowania OSN.

Idea sformułowania zadania tego typu zrodziła się pod wpływem studiów literaturowych z dziedziny projektowania procesów technologicznych [18,39,44,68] oraz automatycznego programowania OSN [10,37,39]. Zadanie to sygnalizuje jedno z możliwych zastosowań przedstawionej w rozprawie metody.

W przykładowym zadaniu występuje prostokątna płytką stalowa o rozmiarach 100x70x20, w której należy wykonać szereg otworów o różnych cechach (rys. 6.2). Zakładamy istnienie potrzebnych narzędzi - fakt ten wyrazimy w opisie zadania użyciem predykatu NARZ(typ narzędzia, średnica). Dysponujemy również odpowiednią OSN - wiertarką z głowicą wielonarzędziową, której repertuar przekształceń technologicznych zawiera wszystkie przekształcenia, niezbędne do zrealizowania procesu technologicznego.

Założmy istnienie następujących sześciu przekształceń technologicznych (przedstawionych od razu w postaci reguł przekształcania $O_1, O_2, \dots, O_6$):

O_1 - mocowanie narzędzia Q1 o średnicy Q2 w pozycji roboczej:

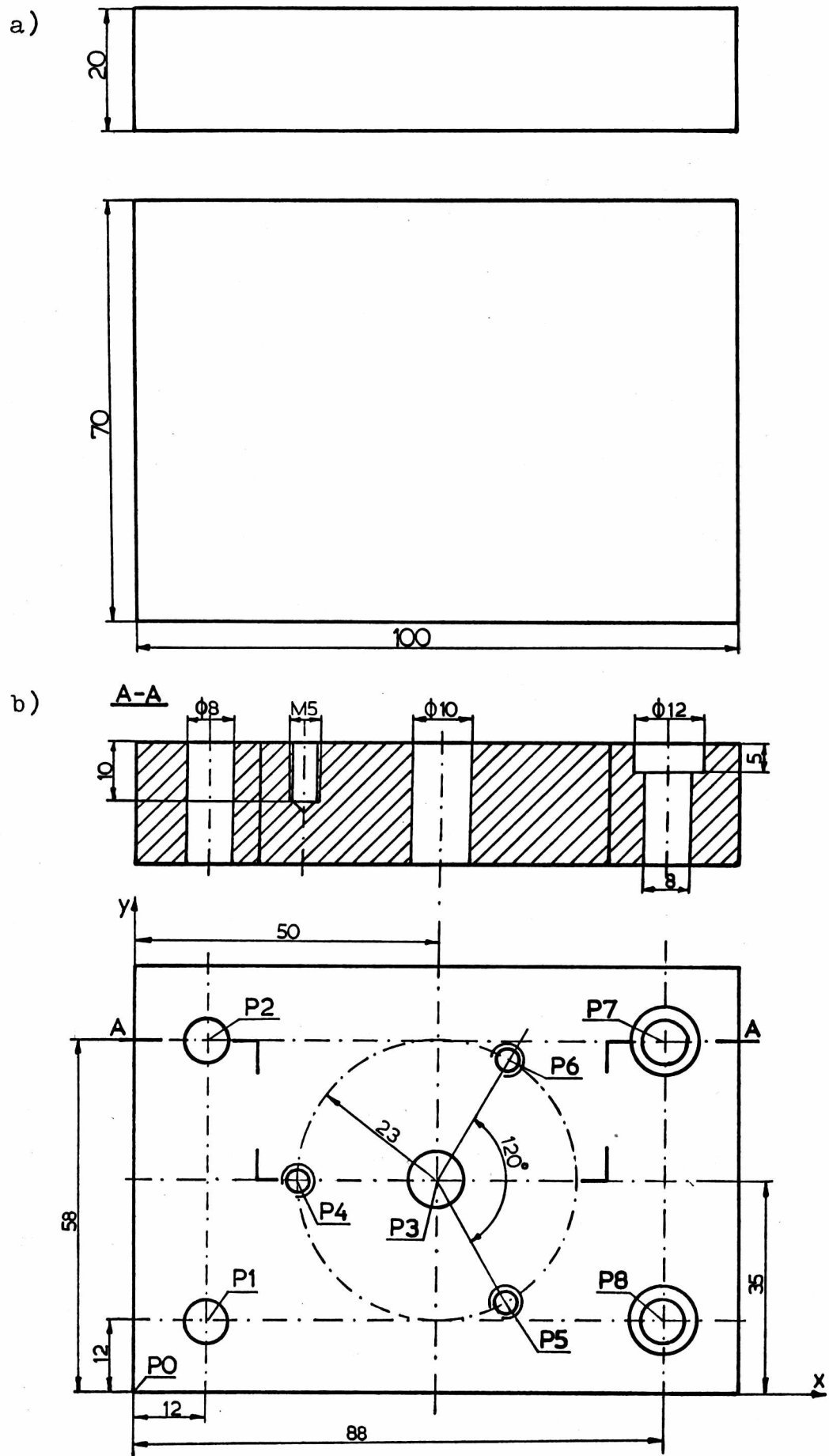
$$Co = \{(co_1) \text{ NARZ}(Q1, Q2)\}$$

$$Ca = \{(ca_1) \text{ ZAMOC}(Q3, Q4)\}$$

$$Ad = \{(ad_1) \text{ ZAMOC}(Q1, Q2)\}$$

$$Re = \emptyset$$

Predykat ZAMOC(Q1,Q2) oznacza: "narzędzie Q1 o średnicy Q2 jest zamocowane w pozycji roboczej".



Rys. 6.2. Płytką z przykładu w p. 6,6 przed (a) i po (b) obróbie

O₂ - przesuw głowicy wiertarki od punktu Q1 do punktu Q2:

$$Co = \{(co_1) \text{ GLOWICA}(Q1)\}$$

$$Ca = \{(ca_1) \text{ GLOWICA}(Q1)\}$$

$$Ad = \{(ad_1) \text{ GLOWICA}(Q2)\}$$

$$Re = \emptyset$$

Predykat GLOWICA(Q1) oznacza: "głowica wiertarki jest ustawiona nad punktem Q1".

O₃ - nawiercanie otworu o średnicy Q2 w punkcie Q1:

$$Co = \{(co_1) \text{ GLOWICA}(Q1), \\ (co_2) \text{ ZAMOC}(\text{NAWIERT}, Q2)\}$$

$$Ca = \emptyset$$

$$Ad = \{(ad_1) \text{ OTWORNW}(Q1, Q2)\}$$

$$Re = \emptyset$$

Predykat OTWORNW(Q1, Q2) oznacza: "w punkcie Q1 istnieje otwór nawiercony nawiertakiem o średnicy Q2".

NAWIERT oznacza narzędzie typu nawiertak.

O₄ - wiercenie otworu o średnicy Q2, na głębokość Q3, w punkcie Q1:

$$Co = \{(co_1) \text{ OTWORNW}(Q1, Q4), \\ (co_2) \text{ GLOWICA}(Q1), \\ (co_3) \text{ ZAMOC}(\text{WIERTLO}, Q2)\}$$

$$Ca = \{(ca_1) \text{ OTWORNW}(Q1, Q4)\}$$

$$Ad = \{(ad_1) \text{ OTWOR}(Q1, Q2, Q3)\}$$

$$Re = \{(re_1) \text{ TAB}(Q2, F4, Q4, F3), \\ (re_2) \text{ TAB}(Q2, F8, Q4, F3), \\ (re_3) \text{ TAB}(Q2, F10, Q4, F3)\}$$

Predykat OTWOR(Q1, Q2, Q3) oznacza: "w punkcie Q1 istnieje otwór o średnicy Q2 i głębokości Q3". F3, F4, F8, F10 oznacza średnice: $\phi 3$, $\phi 4$, $\phi 8$, $\phi 10$. Predykat TAB opisano

w p. 6.3.2.

- O_5 - pogłębianie cylindryczne otworu o średnicy $Q3$ i głębokości $Q4$ w punkcie $Q1$, wiertłem o średnicy $Q2$, na głębokość $Q5$:

$$\begin{aligned}Co &= \{(co_1) \text{ OTWOR}(Q1, Q3, Q4), \\ &\quad (co_2) \text{ GLOWICA}(Q1), \\ &\quad (co_3) \text{ ZAMOC}(\text{WIERTLO}, Q2)\} \\Ca &= \{(ca_1) \text{ OTWOR}(Q1, Q3, Q4)\} \\Ad &= \{(ad_1) \text{ OTWORPC}(Q1, Q3, Q4, Q2, Q5)\} \\Re &= \{(re_1) \text{ MNI}(Q3, Q2), \\ &\quad (re_2) \text{ MNI}(Q5, Q4)\}\end{aligned}$$

Predykat $\text{OTWORPC}(Q1, Q3, Q4, Q2, Q5)$ oznacza: "w punkcie $Q1$ istnieje otwór o średnicy $Q3$ i głębokości $Q4$, pogłębiony cylindrycznie na głębokość $Q5$ do średnicy $Q2$ ". Predykat MNI opisano w p. 6.3.2.

- O_6 - gwintowanie otworu o średnicy $Q3$ i głębokości $Q4$ w punkcie $Q1$, gwintownikiem $Q2$ na głębokość $Q5$:

$$\begin{aligned}Co &= \{(co_1) \text{ OTWOR}(Q1, Q3, Q4), \\ &\quad (co_2) \text{ GLOWICA}(Q1), \\ &\quad (co_3) \text{ ZAMOC}(\text{GWINT}, Q2)\} \\Ca &= \{(ca_1) \text{ OTWOR}(Q1, Q3, Q4)\} \\Ad &= \{(ad_1) \text{ OTWORGW}(Q1, Q2, Q5)\} \\Re &= \{(re_1) \text{ TAB}(Q3, F4, Q2, M5), \\ &\quad (re_2) \text{ TAB}(Q4, G12, Q5, G10), \\ &\quad (re_3) \text{ MNI}(Q5, Q4)\}\end{aligned}$$

Predykat $\text{OTWORGW}(Q1, Q2, Q5)$ oznacza: "w punkcie $Q1$ istnieje otwór gwintowany gwintownikiem $Q2$ na głębokość $Q5$ ".

$G12$ i $G10$ oznaczają głębokości: 12 mm i 10 mm.

Sytuację początkową S_p można opisać następująco:

$$S_p = \{ (f_1) \text{ NARZ}(\text{NAWIERT}, F3), \\ (f_2) \text{ NARZ}(\text{WIERTLO}, F4), \\ (f_3) \text{ NARZ}(\text{WIERTLO}, F8), \\ (f_4) \text{ NARZ}(\text{WIERTLO}, F10), \\ (f_5) \text{ NARZ}(\text{WIERTLO}, F12), \\ (f_6) \text{ NARZ}(\text{GWINT}, M5), \\ (f_7) \text{ GLOWICA}(P0) \}$$

W opisie S_p świadomie pominięto informację o istnieniu płyty stalowej, która ma być obrabiana. Informacja ta nie wnosi bowiem do opisu zadania istotnych treści. M5 w formule f_6 oznacza typ gwintownika GWINT.

Podzbiór S_k sytuacji docelowej S_d wystarczająco precyzyjnie opisują formuły:

$$S_k = \{ (f_1) \text{ OTWOR}(P1, F8, G20), \\ (f_2) \text{ OTWOR}(P2, F8, G20), \\ (f_3) \text{ OTWOR}(P3, F10, G20), \\ (f_4) \text{ OTWORGW}(P4, M5, G10), \\ (f_5) \text{ OTWORGW}(P5, M5, G10), \\ (f_6) \text{ OTWORGW}(P6, M5, G10), \\ (f_7) \text{ OTWORPC}(P7, F8, G20, F12, G5), \\ (f_8) \text{ OTWORPC}(P8, F8, G20, F12, G5) \}$$

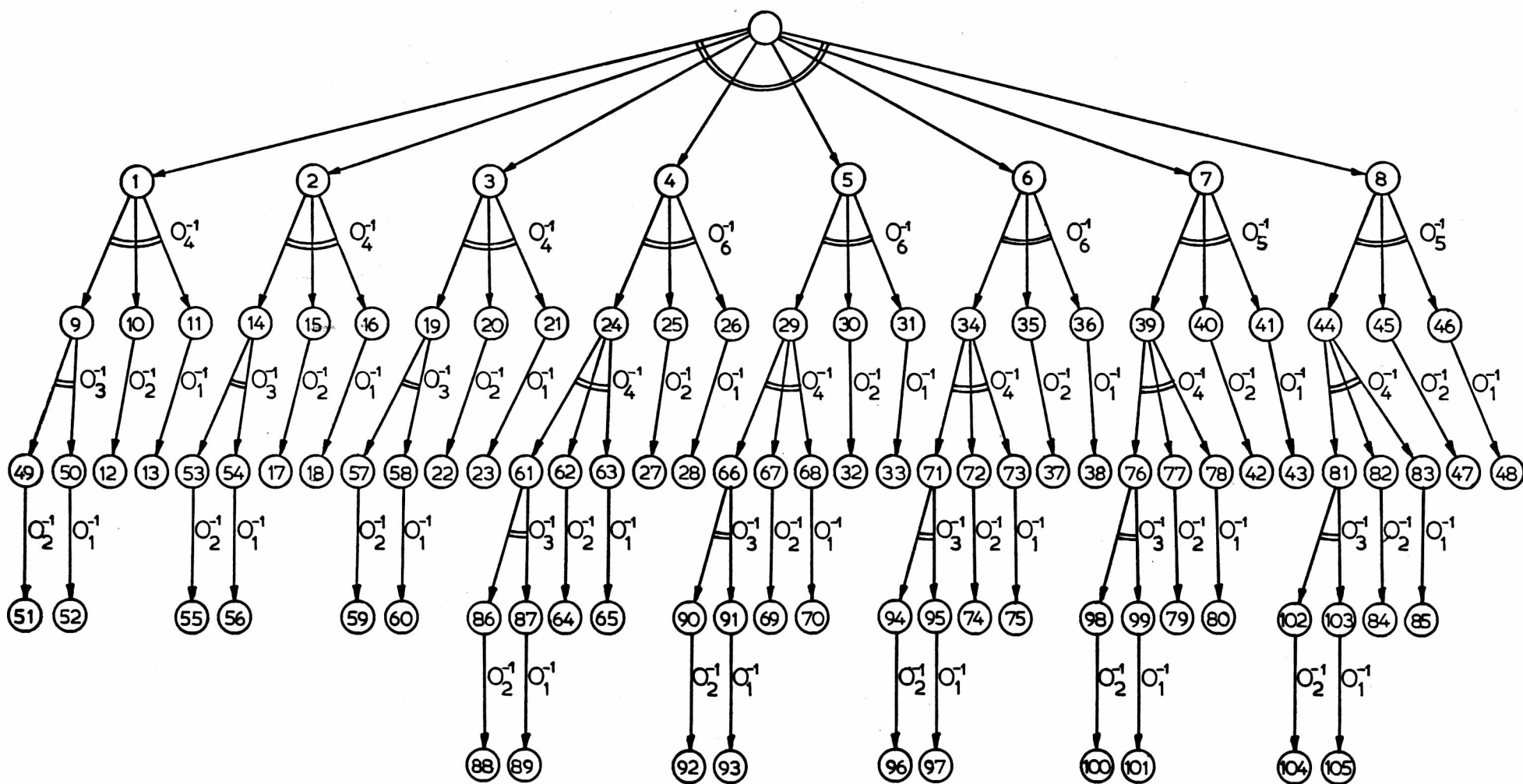
W opisach S_p i S_k występują stałe $P0, P1, P2, \dots, P8$, określające symbolicznie punkty charakterystyczne płaszczyzny. $P0$ określa położenie spoczynkowe głowicy wiertarki, $P1 + P8$ - położenia poszczególnych otworów. Średnice ($\emptyset$) otworów i wiertek oznaczono przez F, głębokości otworów poprzedza litera G, natomiast gwinty oznaczono literą M.

Poniżej zamieszczono tabulogram z przetwarzania konwersacyjnego systemu SPR. Tabulogram ilustruje wprowadzanie do systemu SPR opisu zadania PLYTKA oraz rozwiązywanie tego zadania

przez system w trybie automatycznym, wieloetapowo. W poszczególnych etapach rozwiązywania stosowano różne poziomy protokołowania (niski - 0, średni - 1, wysoki - 2).

Wybrane fragmenty tabulogramu oznaczono symbolami $\ast i$, $i=1, 2, \dots, 12$, w celu umożliwienia odwoływania się do nich w objaśniających opisach, które zamieszczono na końcu tabulogramu.

Rozwiązanie zadania PLYTKA z poziomem protokołowania 1 lub 2 umożliwia odtworzenie całego drzewa podcelów DP, powstającego podczas rozwiązywania tego zadania (rys. 6.3 i tab. 6.3).



Rys. 6.3. Drzewo podcelów dla przykładowego zadania w p. 6.6

Tabela 6.3. Wykaz podcelów drzewa podcelów DP z rys. 6.3

Nr podcelu	Opis podcelu	Nr podcelu	Opis podcelu
1	OTWOR(P1,F8,G20)	30	GLOWICA(P5)
2	OTWOR(P2,F8,G20)	31	ZAMOC(GWINT,M5)
3	OTWOR(P3,F10,G20)	32	GLOWICA(X5)
4	OTWORGW(P4,M5,G10)	33	NARZ(GWINT,M5)
5	OTWORGW(P5,M5,G10)	34	OTWOR(P6,F4,G12)
6	OTWORGW(P6,M5,G10)	35	GLOWICA(P6)
7	OTWORPC(P7,F8,G20,F12,G5)	36	ZAMOC(GWINT,M5)
8	OTWORPC(P8,F8,G20,F12,G5)	37	GLOWICA(X6)
9	OTWORNW(P1,F3)	38	NARZ(GWINT,M5)
10	GLOWICA(P1)	39	OTWOR(P7,F8,G20)
11	ZAMOC(WIERTLO,F8)	40	GLOWICA(P7)
12	GLOWICA(X1)	41	ZAMOC(WIERTLO,F12)
13	NARZ(WIERTLO,F8)	42	GLOWICA(X7)
14	OTWORNW(P2,F3)	43	NARZ(WIERTLO,F12)
15	GLOWICA(P2)	44	OTWOR(P8,F8,G20)
16	ZAMOC(WIERTLO,F8)	45	GLOWICA(P8)
17	GLOWICA(X2)	46	ZAMOC(WIERTLO,F12)
18	NARZ(WIERTLO,F8)	47	GLOWICA(X8)
19	OTWORNW(P3,F3)	48	NARZ(WIERTLO,F12)
20	GLOWICA(P3)	49	GLOWICA(P1)
21	ZAMOC(WIERTLO,F10)	50	ZAMOC(NAWIERT,F3)
22	GLOWICA(X3)	51	GLOWICA(X9)
23	NARZ(WIERTLO,F10)	52	NARZ(NAWIERT,F3)
24	OTWOR(P4,F4,G12)	53	GLOWICA(P2)
25	GLOWICA(P4)	54	ZAMOC(NAWIERT,F3)
26	ZAMOC(GWINT,M5)	55	GLOWICA(X10)
27	GLOWICA(X4)	56	NARZ(NAWIERT,F3)
28	NARZ(GWINT,M5)	57	GLOWICA(P3)
29	OTWOR(P5,F4,G12)	58	ZAMOC(NAWIERT,F3)

Tabela 6.3 - c.d.

Nr podcelu	Opis podcelu	Nr podcelu	Opis podcelu
59	GLOWICA(X11)	83	ZAMOC(WIERTLO,F8)
60	NARZ(NAWIERT,F3)	84	GLOWICA(X16)
61	OTWORNW(P4,F3)	85	NARZ(WIERTLO,F8)
62	GLOWICA(P4)	86	GLOWICA(P4)
63	ZAMOC(WIERTLO,F4)	87	ZAMOC(NAWIERT,F3)
64	GLOWICA(X12)	88	GLOWICA(X17)
65	NARZ(WIERTLO,F4)	89	NARZ(NAWIERT,F3)
66	OTWORNW(P5,F3)	90	GLOWICA(P5)
67	GLOWICA(P5)	91	ZAMOC(NAWIERT,F3)
68	ZAMOC(WIERTLO,F4)	92	GLOWICA(X18)
69	GLOWICA(X13)	93	NARZ(NAWIERT,F3)
70	NARZ(WIERTLO,F4)	94	GLOWICA(P6)
71	OTWORNW(P6,F3)	95	ZAMOC(NAWIERT,F3)
72	GLOWICA(P6)	96	GLOWICA(X19)
73	ZAMOC(WIERTLO,F4)	97	NARZ(NAWIERT,F3)
74	GLOWICA(X14)	98	GLOWICA(P7)
75	NARZ(WIERTLO,F4)	99	ZAMOC(NAWIERT,F3)
76	OTWORNW(P7,F3)	100	GLOWICA(X20)
77	GLOWICA(P7)	101	NARZ(NAWIERT,F3)
78	ZAMOC(WIERTLO,F8)	102	GLOWICA(P8)
79	GLOWICA(X15)	103	ZAMOC(NAWIERT,F3)
80	NARZ(WIERTLO,F8)	104	GLOWICA(X21)
81	OTWORNW(P8,F3)	105	NARZ(NAWIERT,F3)
82	GLOWICA(P8)		

11.32.35_ SPR FR

=====

====< SYSTEM POSZUKIWANIA ROZWIAZAN SPR D:20/09/83 T:11/33/02

=< REALIZATOR SPR

```
- :LOGIN DIAL-INT-001
=? HASLO
- ZS
- :PREDICAT NARZ,GLOWICA,OTWOR,OTWORGW,OTWORPC,ZAMOC,OTWORNW
- :PREDICAT BRAK
- :CONSTANT NAWIERT,WIERTLO,GWINT,F3,F4,F8,F10,F12,F5,M5
- :CONSTANT P0,P1,P2,P3,P4,P5,P6,P7,P8,G20,G10,G5,G12
- :INITIAL START
- :FORMULA NARZ(NAWIERT,F3)
  (F1) NARZ(NAWIERT,F3)
- :FM NARZ(WIERTLO,F4)
  (F2) NARZ(WIERTLO,F4)
- :FM NARZ(WIERTLO,F8)
  (F3) NARZ(WIERTLO,F8)
- :FM NARZ(WIERTLO,F10)
  (F4) NARZ(WIERTLO,F10)
- :FM NARZ(WIERTLO,F12)
  (F5) NARZ(WIERTLO,F12)
- :FM NARZ(GWINT,M5)
  (F6) NARZ(GWINT,M5)
- :FM GLOWICA(P0)
  (F7) GLOWICA(P0)
- :END IT
- :GOAL STOP
- :FM OTWOR(P1,F8,G20)
  (F8) OTWOR(P1,F8,G20)
- :FM OTWOR(P2,F8,G20)
  (F9) OTWOR(P2,F8,G20)
- :FM OTWOR(P3,F10,G20)
  (F10) OTWOR(P3,F10,G20)
- :FM OTWORGW(P4,M5,G10)
  (F11) OTWORGW(P4,M5,G10)
- :FM OTWORGW(P5,M5,G10)
  (F12) OTWORGW(P5,M5,G10)
- :FM OTWORGW(P6,M5,G10)
  (F13) OTWORGW(P6,M5,G10)
- :FM OTWORPC(P7,F8,G20,F12,G5)
  (F14) OTWORPC(P7,F8,G20,F12,G5)
- :FM OTWORPC(P8,F8,G20,F12,G5)
  (F15) OTWORPC(P8,F8,G20,F12,G5)
- :END GL
- :OPERATOR MOCOW
- :COND
- :FM NARZ(Q1,Q2)
  (F16) NARZ(Q1,Q2)
- :END CO
- :CANC
- :FM ZAMOC(Q3,Q4)
```

*1

*2

*3

*4

*5

```
(F17) ZAMOC(Q3,Q4)
- #END CA
- #ADD
- #FM ZAMOC(Q1,Q2)
(F18) ZAMOC(Q1,Q2)
- #END AD
- #END OP

- #OP PRZESUW
- #CO
- #FM GLOWICA(Q1)
(F19) GLOWICA(Q1)
- #END CO
- #CA
- *19
- #END CA
- #AD
- #FM GLOWICA(Q2)
(F20) GLOWICA(Q2)
- #END AD
- #END OP

- #OP NAWIERC
- #CO
- *19
- #FM ZAMOC(NAWIERT,Q2)
(F21) ZAMOC(NAWIERT,Q2)
- #END CO
- #CA
- #FM BRAK
(F22) BRAK
- #END CA
- #AD
- #FM OTWORNW(Q1,Q2)
(F23) OTWORNW(Q1,Q2)
- #END AD
- #END OP

- #OP WIERC
- #CO
- *19
- #FM OTWORNW(Q1,Q4)
(F24) OTWORNW(Q1,Q4)
- #FM ZAMOC(WIERTLO,Q2)
(F25) ZAMOC(WIERTLO,Q2)
- #END CO
- #CA
- *24
- #END CA
- #AD
- #FM OTWOR(Q1,Q2,Q3)
(F26) OTWOR(Q1,Q2,Q3)
- #END AD
- #RE
- #FM TAB(Q2,F4,Q4,F3)
(F27) TAB(Q2,F4,Q4,F3)
- #FM TAB(Q2,F8,Q4,F3)
(F28) TAB(Q2,F8,Q4,F3)
- #FM TAB(Q2,F10,Q4,F3)
(F29) TAB(Q2,F10,Q4,F3)
```

```

- :END RE
- :END OP

- :OP POGL
- :CO
- :FM OTWOR(Q1,Q3,Q4)
- (F30) OTWOR(Q1,Q3,Q4)
- *19,*25
- :END CO
- :CA
- *30
- :END CA
- :AD
- :FM OTWORPC(Q1,Q3,Q4,Q2,Q5)
- (F31) OTWORPC(Q1,Q3,Q4,Q2,Q5)
- :END AD
- :RE
- :FM MNI(Q3,Q2)
- (F32) MNI(Q3,Q2)
- :FM MNI(Q5,Q4)
- (F33) MNI(Q5,Q4)
- :END RE
- :END OP

- :OP GWINTOW
- :CO
- *30,*19
- :FM ZAMOC(GWINT,Q2)
- (F34) ZAMOC(GWINT,Q2)
- :END CO
- :CA
- *30
- :END CA
- :AD
- :FM OTWORGW(Q1,Q2,Q5)
- (F35) OTWORGW(Q1,Q2,Q5)
- :END AD
- :RE
- :FM TAB(Q3,F4,Q2,M5)
- (F36) TAB(Q3,F4,Q2,M5)
- :FM TAB(Q4,G12,Q5,G10)
- (F37) TAB(Q4,G12,Q5,G10)
- :FM MNI(Q5,Q4)
- (F38) MNI(Q5,Q4)
- :END RE
- :END OP

- :EXERCISE PLYTKA
- :INITIAL *START
- :GOAL *STOP
- :OPERATOR *MOCOW,*PRZESUW,*NAWIERC,*WIERC,*POGL,*GWINTOW
- :END EX

- :STRATEGY C
- :FUNCTION G

```

```

LAMBDA 1 = 1.000
LAMBDA 2 = 1.000
LAMBDA 3 = 1.000
LAMBDA 4 = 1.000

```

*6

*7

LAMBDA 5 = 1.000
LAMBDA 6 = 1.000

- :LIMIT 600
- :REPORT 2
- :DO PLYTKA

*8

==< PROCEDURA POSZUKIWANIA ROZWIĄZANIA D:20/09/83 T:12/24/37
==< POZIOM RAPORTOWANIA: 2
==< ZADANIE: PLYTKA

=< SYTUACJA POCZĄTKOWA - "S1"

(E1) NARZ(NAWIERT,F3)
(E2) NARZ(WIERTLO,F4)
(E3) NARZ(WIERTLO,F8)
(E4) NARZ(WIERTLO,F10)
(E5) NARZ(WIERTLO,F12)
(E6) NARZ(GWINT,M5)
(E7) GŁOWICA(P0)

*9

=< CEL:

(P1) OTWOR(P1,F8,G20)
(P2) OTWOR(P2,F8,G20)
(P3) OTWOR(P3,F10,G20)
(P4) OTWORGW(P4,M5,G10)
(P5) OTWORGW(P5,M5,G10)
(P6) OTWORGW(P6,M5,G10)
(P7) OTWORPC(P7,F8,G20,F12,G5)
(P8) OTWORPC(P8,F8,G20,F12,G5)

=< REGUŁY PRZESZTALCANIA:

(01) MOCOW
(02) PRZESUW
(03) NAWIERC
(04) WIERC
(05) POGL
(06) GWINTOW

- :RETURN 2
- :GO

=< KROK 1

"P1" ROZWINIĘTO WSTECZNIE ZA POMOCĄ REGUŁY 04:"WIERC"
UTWORZONO PODCELE:

(P9) OTWORNW(P1,F3)
(P10) GŁOWICA(P1)
(P11) ZAMOC(WIERTLO,F8)

ANALIZA PRZERWANA BEZ OSIĄGNIĘCIA CELU

=< KROK 2

"P10" ROZWINIĘTO WSTECZNIE ZA POMOCĄ REGUŁY 02:"PRZESUW"
UTWORZONO PODCELE:

(P12) GŁOWICA(X1)

SYTUACJE "S1" PRZEKSZTAŁCONO ZA POMOCĄ REGUŁY 02: "PRZESUN"
UTWORZONO SYTUACJE "S2"

WYRAŻENIA WYKRESŁONE:

(E7) GŁOWICA(P0)

WYRAŻENIA DOLĄCZONE:

(E8) GŁOWICA(P1)

S2:

(E1) NARZ(NAWIERT,F3)
(E2) NARZ(WIERTLO,F4)
(E3) NARZ(WIERTLO,F8)
(E4) NARZ(WIERTLO,F10)
(E5) NARZ(WIERTLO,F12)
(E6) NARZ(GWINT,M5)
(E8) GŁOWICA(P1)

ANALIZA PRZERWANA BEZ OSIĄGNIĘCIA CELU

=< WYKONANO 2. KROK W POSZUKIWANIU ROZWIĄZANIA

- :RP 1
- :RT 1
- :GO

=< KROK 3

"F11" ----< 01:"MOCOW"

(F13) NARZ(WIERTLO,F8)

S1 ----> 01:"MOCOW" ----> S3

AD:

(E9) ZAMOC(WIERTLO,F8)

S3:

(E1) (E2) (E3) (E4) (E5) (E6) (E7) (E9)

ANALIZA PRZERWANA BEZ OSIĄGNIĘCIA CELU

=< WYKONANO 3. KROK W POSZUKIWANIU ROZWIĄZANIA

- :RP 0
- :RT 23
- :GO

=< WYKONANO 26. KROK W POSZUKIWANIU ROZWIĄZANIA

- :RT 1
- :RP 2
- :GO

=< KROK 27

"P50" ROZWINIETO WSTECZNIE ZA POMOCĄ REGUŁY 01: "MOCOW"
UTWORZONO PODCELE:

(P52) NARZ(NAWIERT,F3)

SYTUACJE "S1" PRZEKSZTAŁCONO ZA POMOCĄ REGUŁY 01: "MOCOW"
UTWORZONO SYTUACJE "S19"

WYRAZENIA DOLACZONE:

(E25) ZAMOC(NAWIERT,F3)

S19:

(E1) NARZ(NAWIERT,F3)
(E2) NARZ(WIERTLO,F4)
(E3) NARZ(WIERTLO,F8)
(E4) NARZ(WIERTLO,F10)
(E5) NARZ(WIERTLO,F12)
(E6) NARZ(GWINT,M5)
(E7) GLOWICA(P0)
(E25) ZAMOC(NAWIERT,F3)

SYTUACJE "S19" PRZEKSZTAŁCONO ZA POMOCĄ REGUŁY 02: "PRZESUW"
UTWORZONO SYTUACJE "S20"

WYRAZENIA WYKRESŁONE:

(E7) GLOWICA(P0)

WYRAZENIA DOLACZONE:

(E26) GLOWICA(P1)

S20:

(E1) NARZ(NAWIERT,F3)
(E2) NARZ(WIERTLO,F4)
(E3) NARZ(WIERTLO,F8)
(E4) NARZ(WIERTLO,F10)
(E5) NARZ(WIERTLO,F12)
(E6) NARZ(GWINT,M5)
(E25) ZAMOC(NAWIERT,F3)
(E26) GLOWICA(P1)

SYTUACJE "S20" PRZEKSZTAŁCONO ZA POMOCĄ REGUŁY 03: "NAWIERC"
UTWORZONO SYTUACJE "S21"

WYRAZENIA DOLACZONE:

(E27) OTWORNW(P1,F3)

S21:

(E1) NARZ(NAWIERT,F3)
(E2) NARZ(WIERTLO,F4)
(E3) NARZ(WIERTLO,F8)
(E4) NARZ(WIERTLO,F10)
(E5) NARZ(WIERTLO,F12)
(E6) NARZ(GWINT,M5)

(E25) ZAMOC(NAWIERT,F3)
(E26) GLOWICA(P1)
(E27) OTWORNW(P1,F3)

SYTUACJE "S21" PRZEKSZTALCONO ZA POMOCĄ REGUŁY 02:"MOCOW"
UTWORZONO SYTUACJE "S22"

WYRAZENIA WYKRESŁONE:

(E25) ZAMOC(NAWIERT,F3)

WYRAZENIA DOLACZONE:

(E28) ZAMOC(WIERTLO,F8)

S22:

(E1) NARZ(NAWIERT,F3)
(E2) NARZ(WIERTLO,F4)
(E3) NARZ(WIERTLO,F8)
(E4) NARZ(WIERTLO,F10)
(E5) NARZ(WIERTLO,F12)
(E6) NARZ(GWINT,M5)
(E26) GLOWICA(P1)
(E27) OTWORNW(P1,F3)
(E28) ZAMOC(WIERTLO,F8)

SYTUACJE "S22" PRZEKSZTALCONO ZA POMOCĄ REGUŁY 04:"WIERC"
UTWORZONO SYTUACJE "S23"

WYRAZENIA WYKRESŁONE:

(E27) OTWORNW(P1,F3)

WYRAZENIA DOLACZONE:

(E29) OTWOR(P1,F8,G20)

S23:

(E1) NARZ(NAWIERT,F3)
(E2) NARZ(WIERTLO,F4)
(E3) NARZ(WIERTLO,F8)
(E4) NARZ(WIERTLO,F10)
(E5) NARZ(WIERTLO,F12)
(E6) NARZ(GWINT,M5)
(E26) GLOWICA(P1)
(E28) ZAMOC(WIERTLO,F8)
(E29) OTWOR(P1,F8,G20)

ANALIZA PRZERWANA BEZ OSIĄGNIĘCIA CELU

=< WYKONANO 27. KROK W POSZUKIWANIU ROZWIĄZANIA

_ :RP 1
_ :RT 3
_ :GO

=< KROK 28

"P14" ----< 03:"NAWIERC"

(P53) GLOWICA(P2)

(P54) ZAMOC(NAWIERT,F3)

ANALIZA PRZERWANA BEZ OSIAGNIECIA CELU

=< KROK 29

"P53" ----< 02:"PRZESUW"

(P55) GLOWICA(X10)

S23 ----> 02:"PRZESUW" ----> S24

CA:

(E26) GLOWICA(P1)

AD:

(E30) GLOWICA(P2)

S24:

(E1) (E2) (E3) (E4) (E5) (E6) (E28) (E29) (E30)

ANALIZA PRZERWANA BEZ OSIAGNIECIA CELU

=< KROK 30

"P54" ----< 01:"MOCOW"

(P56) NARZ(NAWIERT,F3)

S23 ----> 01:"MOCOW" ----> S25

CA:

(E28) ZAMOC(WIERTLO,F8)

AD:

(E31) ZAMOC(NAWIERT,F3)

S25:

(E1) (E2) (E3) (E4) (E5) (E6) (E28) (E29) (E31)

S25 ----> 02:"PRZESUW" ----> S26

CA:

(E26) GLOWICA(P1)

AD:

(E32) GLOWICA(P2)

S26:

(E1) (E2) (E3) (E4) (E5) (E6) (E29) (E31) (E32)

S26 ----> 03:"NAWIERC" ----> S27

AD:

(E33) OTWORNW(P2,F3)

S27:

(E1) (E2) (E3) (E4) (E5) (E6) (E29) (E31) (E32) (E33)

S27 ----> 01: "MOCOW" ----> S28

CA:

(E31) ZAMOC (NAWIERT, F3)

AD:

(E34) ZAMOC (WIERTLO, F8)

S28:

(E1) (E2) (E3) (E4) (E5)

(E6) (E29) (E32) (E33) (E34)

S28 ----> 04: "WIERC" ----> S29

CA:

(E33) OTWORNW (P2, F3)

AD:

(E35) OTWOR (P2, F8, G20)

S29:

(E1) (E2) (E3) (E4) (E5) (E6) (E29) (E32) (E34) (E35)

ANALIZA PRZERWANA BEZ OSIAGNIECIA CELU

=< WYKONANO 30. KROK W POSZUKIWANIU ROZWIAZANIA

_ :RP 0

_ :GO

=< OSIAGNIETO ROZWIAZANIE W SYTUACJI "S80"

_ :RP 1

_ :SQ S80

=< SEKWENCJA PRZESZTALCAJACA S1 ---->, ... ----> S80:

S1: (E1) NARZ (NAWIERT, F3)

(E2) NARZ (WIERTLO, F4)

(E3) NARZ (WIERTLO, F8)

(E4) NARZ (WIERTLO, F10)

(E5) NARZ (WIERTLO, F12)

(E6) NARZ (GWINT, M5)

(E7) GLOWICA (P0)

S1 ----> 01: "MOCOW" ----> S19

S19 = S1 U AD

AD: (E25) ZAMOC (NAWIERT, F3)

S19: (E1) (E2) (E3) (E4) (E5) (E6) (E7) (E25)

S19 ----> 02: "PRZESUW" ----> S20

S20 = S19 \ CA U AD

CA: (E7)

AD: (E26) GLOWICA (P1)

S20: (E1) (E2) (E3) (E4) (E5) (E6) (E25) (E26)

|
*10
|

|
*11
|
:
|

S2^ ----> 03: "NAWIERC" ----> S21

S21 = S20 U AD
AD: (E27) OTWORNW(P1,F3)
S21: (E1) (E2) (E3) (E4) (E5) (E6) (E25) (E26) (E27)

S21 ----> 01: "MOCOW" ----> S22

S22 = S21 \ CA U AD
CA: (E25)
AD: (E28) ZAMOC(WIERTLO,F8)
S22: (E1) (E2) (E3) (E4) (E5) (E6) (E26) (E27) (E28)

S22 ----> 04: "WIERC" ----> S23

S23 = S22 \ CA U AD
CA: (E27)
AD: (E29) OTWOR(P1,F8,G20)
S23: (E1) (E2) (E3) (E4) (E5) (E6) (E26) (E28) (E29)

S23 ----> 01: "MOCOW" ----> S25

S25 = S23 \ CA U AD
CA: (E28)
AD: (E31) ZAMOC(NAWIERT,F3)
S25: (E1) (E2) (E3) (E4) (E5) (E6) (E26) (E29) (E31)

S25 ----> 02: "PRZESUW" ----> S26

S26 = S25 \ CA U AD
CA: (E26)
AD: (E32) GLOWICA(P2)
S26: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E31) (E32)

S26 ----> 03: "NAWIERC" ----> S27

S27 = S26 U AD
AD: (E33) OTWORNW(P2,F3)
S27: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E31) (E32) (E33)

S27 ----> 01: "MOCOW" ----> S28

S28 = S27 \ CA U AD
CA: (E31)
AD: (E34) ZAMOC(WIERTLO,F8)
S28: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E32) (E33) (E34)

S28 ----> 04: "WIERC" ----> S29

S29 = S28 \ CA U AD
CA: (E33)
AD: (E35) OTWOR(P2,F8,G20)
S29: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E32) (E34) (E35)

S29 ----> 01: "MOCOW" ----> S31

S31 = S29 \ CA U AD
CA: (E34)
AD: (E37) ZAMOC(NAWIER1,F3)
S31: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E32) (E35) (E37)

S31 ----> 02: "PRZESUW" ----> S32

S32 = S31 \ CA U AD

CA: (E32)

AD: (E38) GLOWICA(P3)

S32: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E37) (E38)

S32 ----> 03: "NAWIERC" ----> S33

S33 = S32 U AD

AD: (E39) OTWORNW(P3,F3)

S33: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E37) (E38)
(E39)

S33 ----> 01: "MOCOW" ----> S34

S34 = S33 \ CA U AD

CA: (E37)

AD: (E40) ZAMOC(WIERTLO,F10)

S34: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E38) (E39)
(E40)

S34 ----> 04: "WIERC" ----> S35

S35 = S34 \ CA U AD

CA: (E39)

AD: (E41) OTWOR(P3,F10,G20)

S35: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E38) (E40)
(E41)

S35 ----> 01: "MOCOW" ----> S42

S42 = S35 \ CA U AD

CA: (E40)

AD: (E48) ZAMOC(NAWIERT,F3)

S42: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E38) (E41)
(E48)

S42 ----> 02: "PRZESUW" ----> S43

S43 = S42 \ CA U AD

CA: (E38)

AD: (E49) GLOWICA(P4)

S43: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E48)
(E49)

S43 ----> 03: "NAWIERC" ----> S44

S44 = S43 U AD

AD: (E50) OTWORNW(P4,F3)

S44: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E48)
(E49) (E50)

S44 ----> 01: "MOCOW" ----> S45

S45 = S44 \ CA U AD

CA: (E48)

AD: (E51) ZAMOC(WIERTLO,F4)

S45: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E49)
(E50) (E51)

S45 ----> 04: "WIERC" ----> S46

S46 = S45 \ CA U AD

CA: (E50)

AD: (E52) OTWOR(P4,F4,G12)

S46: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E49)
(E51) (E52)

S46 ----> 01: "MOCOW" ----> S47

S47 = S46 \ CA U AD

CA: (E51)

AD: (E53) ZAMOC(GWINT,M5)

S47: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E49)
(E52) (E53)

S47 ----> 06: "GWINTOW" ----> S48

S48 = S47 \ CA U AD

CA: (E52)

AD: (E54) OTWORGW(P4,M5,G10)

S48: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E49)
(E53) (E54)

S48 ----> 01: "MOCOW" ----> S50

S50 = S48 \ CA U AD

CA: (E53)

AD: (E56) ZAMOC(NAWIERT,F3)

S50: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E49)
(E54) (E56)

S50 ----> 02: "PRZESUW" ----> S51

S51 = S50 \ CA U AD

CA: (E49)

AD: (E57) GLOWICA(P5)

S51: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E54)
(E56) (E57)

S51 ----> 03: "NAWIERC" ----> S52

S52 = S51 U AD

AD: (E58) OTWORNW(P5,F3)

S52: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E54)
(E56) (E57) (E58)

S52 ----> 01: "MOCOW" ----> S53

S53 = S52 \ CA U AD

CA: (E56)

AD: (E59) ZAMOC(WIERTLO,F4)

S53: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E54)
(E57) (E58) (E59)

S53 ----> 04: "WIERC" ----> S54

S54 = S53 \ CA U AD

CA: (E58)

AD: (E60) OTWOR(P5,F4,G12)

S54: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E54)
(E57) (E59) (E60)

S54 ----> 01: "MOCOW" ----> S55

S55 = S54 \ CA U AD

CA: (E59)

AD: (E61) ZAMOC(GWINT, M5)

S55: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E54)
(E57) (E60) (E61)

S55 ----> 06: "GWINTOW" ----> S56

S56 = S55 \ CA U AD

CA: (E60)

AD: (E62) OTWORGW(P5, M5, G10)

S56: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E54)
(E57) (E61) (E62)

S56 ----> 01: "MOCOW" ----> S58

S58 = S56 \ CA U AD

CA: (E61)

AD: (E64) ZAMOC(NAWIERT, F3)

S58: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E54)
(E57) (E62) (E64)

S58 ----> 02: "PRZESUW" ----> S59

S59 = S58 \ CA U AD

CA: (E57)

AD: (E65) GLOWICA(P6)

S59: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E54)
(E62) (E64) (E65)

S59 ----> 03: "NAWIERC" ----> S60

S60 = S59 U AD

AD: (E66) OTWORNW(P6, F3)

S60: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E54)
(E62) (E64) (E65) (E66)

S60 ----> 01: "MOCOW" ----> S61

S61 = S60 \ CA U AD

CA: (E64)

AD: (E67) ZAMOC(WIERTLO, F4)

S61: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E54)
(E62) (E65) (E66) (E67)

S61 ----> 04: "WIERC" ----> S62

S62 = S61 \ CA U AD

CA: (E66)

AD: (E68) OTWOR(P6, F4, G12)

S62: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E54)
(E62) (E65) (E67) (E68)

S62 ----> 01: "MOCOW" ----> S63

S63 = S62 \ CA U AD
CA: (E67)
AD: (E69) ZAMOC(GWINT,M5)
S63: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E54)
(E62) (E65) (E68) (E69)

S63 ----> 06: "GWINTOW" ----> S64

S64 = S63 \ CA U AD
CA: (E68)
AD: (E70) OTWOROW(P6,M5,G10)
S64: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E54)
(E62) (E65) (E69) (E70)

S64 ----> 01: "MOCOW" ----> S66

S66 = S64 \ CA U AD
CA: (E69)
AD: (E72) ZAMOC(NAWIERT,F3)
S66: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E54)
(E62) (E65) (E70) (E72)

S66 ----> 02: "PRZESUW" ----> S67

S67 = S66 \ CA U AD
CA: (E65)
AD: (E73) GLOWICA(P7)
S67: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E54)
(E62) (E70) (E72) (E73)

S67 ----> 03: "NAWIERC" ----> S68

S68 = S67 U AD
AD: (E74) OTWORNW(P7,F3)
S68: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E54)
(E62) (E70) (E72) (E73) (E74)

S68 ----> 01: "MOCOW" ----> S69

S69 = S68 \ CA U AD
CA: (E72)
AD: (E75) ZAMOC(WIERTLO,F8)
S69: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E54)
(E62) (E70) (E73) (E74) (E75)

S69 ----> 04: "WIERC" ----> S70

S70 = S69 \ CA U AD
CA: (E74)
AD: (E76) OTWOR(P7,F8,G20)
S70: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E54)
(E62) (E70) (E73) (E75) (E76)

S70 ----> 01: "MOCOW" ----> S71

S71 = S70 \ CA U AD
CA: (E75)
AD: (E77) ZAMOC(WIERTLO,F12)
S71: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E54)
(E62) (E70) (E73) (E76) (E77)

S71 ----> 04: "WIERC" ----> S72

S72 = S71 \ CA U AD

CA: (E76)

AD: (E78) OTWORPC(P7,F8,G20,F12,G5)

S72: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E54)
(E62) (E70) (E73) (E77) (E78)

S72 ----> 01: "MOCOW" ----> S74

S74 = S72 \ CA U AD

CA: (E77)

AD: (E80) ZAMOC(NAWIERT,F3)

S74: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E54)
(E62) (E70) (E73) (E78) (E80)

S74 ----> 02: "PRZESUW" ----> S75

S75 = S74 \ CA U AD

CA: (E73)

AD: (E81) GLOWICA(P8)

S75: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E54)
(E62) (E70) (E78) (E80) (E81)

S75 ----> 03: "NAWIERC" ----> S76

S76 = S75 U AD

AD: (E82) OTWORNW(P8,F3)

S76: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E54)
(E62) (E70) (E78) (E80) (E81) (E82)

S76 ----> 01: "MOCOW" ----> S77

S77 = S76 \ CA U AD

CA: (E80)

AD: (E83) ZAMOC(WIERTLO,F8)

S77: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E54)
(E62) (E70) (E78) (E81) (E82) (E83)

S77 ----> 04: "WIERC" ----> S78

S78 = S77 \ CA U AD

CA: (E82)

AD: (E84) OTWOR(P8,F8,G20)

S78: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E54)
(E62) (E70) (E78) (E81) (E83) (E84)

S78 ----> 01: "MOCOW" ----> S79

S79 = S78 \ CA U AD

CA: (E83)

AD: (E85) ZAMOC(WIERTLO,F12)

S79: (E1) (E2) (E3) (E4) (E5) (E6) (E29) (E35) (E41) (E54)
(E62) (E70) (E78) (E81) (E84) (E85)

S79 ----> 04: "WIERC" ----> S80

S80 = S79 \ CA U AD

CA: (E84)

AD: (E86) OTWORPC(P8,F8,G20,F12,G5)

```
S80: (E1) NARZ (NAWIERT, F3)
      (E2) NARZ (WIERILO, F4)
      (E3) NARZ (WIERILO, F8)
      (E4) NARZ (WIERILO, F10)
      (E5) NARZ (WIERILO, F12)
      (E6) NARZ (GWINT, M5)
      (E29) UTWOR (P1, F8, G20)
      (E35) UTWOR (P2, F8, G20)
      (E41) UTWOR (P3, F10, G20)
      (E54) UTWORGW (P4, M5, G10)
      (E62) UTWORGW (P5, M5, G10)
      (E70) UTWORGW (P6, M5, G10)
      (E78) UTWORFC (P7, F8, G20, F12, G5)
      (E81) GLOWICA (P8)
      (E85) ZAMOC (WIERILO, F12)
      (E86) UTWORFC (P8, F8, G20, F12, G5)
```

```
_ :LOGOUT
_ :FINISH
```

————— *12

=< KONIEC PRZEBIEGU MODULU REALIZATOR SPR

=====

END OF MACRO
13.48.07...

Oto opis poszczególnych fragmentów tabulogramu:

- *1 Inicjacja zadania obliczeniowego przez użytkownika o nazwie DIAL-INT-001 (hasło ZS).
- *2 Deklaracja nazw predykatów i nazw stałych, które występują w zadaniu problemowym.
- *3 Wprowadzanie opisu sytuacji początkowej o nazwie START.
- *4 Wprowadzanie opisu sytuacji docelowej o nazwie STOP.
- *5 Wprowadzanie opisu reguły przekształcania HOCOW oraz pozostałych reguł przekształcania.
- *6 Utworzenie opisu zadania problemowego PLYTKA za pomocą wprowadzonych wcześniej opisów: sytuacji początkowej START, sytuacji docelowej STOP oraz reguł przekształcania.
- *7 Wybór strategii, według której będzie poszukiwana sekwencja przekształcająca (wariant C - strategia uporządkowanego wyboru).
- *8 Zainicjowanie rozwiązywania zadania PLYTKA (uruchomienie procedury poszukiwania rozwiązania).

- *9 Protokół z przebiegu procesu wieloetapowego rozwiązywania zadania PLYTKA, z różnymi poziomami protokołowania w poszczególnych etapach procesu.
- *10 Ustalenie niskiego poziomu protokołowania (0) w poszukiwaniu rozwiązania (w celu uniknięcia dużej liczby wydruków), informacja o osiągnięciu przez procedurę sytuacji S80, zawierającej cel.
- *11 Wyprowadzenie znalezionej sekwencji przekształcającej ze średnim poziomem protokołowania (1).
- *12 Zakończenie zadania obliczeniowego, zakończenie przetwarzania.

Po uzyskaniu rozwiązania zadania PLYTKA przeprowadzono modyfikację funkcji oceny g , w oparciu o znaną sekwencję przekształcającą $S1 \rightarrow \dots \rightarrow S80$. Poniżej zamieszczono tabulogram z przetwarzania konwersacyjnego modułu MODYFIKATOR systemu SPR. Efektem działania procedury modyfikacji funkcji oceny są nowe wartości współczynników λ_i , $i=1,2,\dots,6$.

14.12.53_ SPR FM

=====

===< SYSTEM POSZUKIWANIA ROZWIĄZAN SPR D:21/09/83 T:14/13/41

=< MODYFIKATOR SPR

_ :LOGIN DIAL-INT-001

=? HASLO

_ ZS

_ :MODIFY *PLYTKA,S80

===< PROCEDURA MODYFIKACJI FUNKCJI OCENY D:21/09/83 T:14/14/57

===< ZADANIE: PLYTKA

===< SEKWENCJA PRZEKSZTAŁCAJĄCA: S1 ---> ... ---> S80

=< DOTYCHCZASOWE WARTOSCI WSPOLCZYNNIKOW WAGOWYCH FUNKCJI G:

LAMBDA 1 = 1.000

LAMBDA 2 = 1.000

LAMBDA 3 = 1.000

LAMBDA 4 = 1.000

LAMBDA 5 = 1.000

LAMBDA 6 = 1.000

=< ZMODYFIKOWANE WARTOSCI WSPOLCZYNNIKOW WAGOWYCH FUNKCJI G:

LAMBDA 1 = 0.504

LAMBDA 2 = 0.500

LAMBDA 3 = 0.973

LAMBDA 4 = 0.905

LAMBDA 5 = 1.000

LAMBDA 6 = 0.818

=< KONIEC PRZEBIEGU PROCEDURY MODYFIKACJI FUNKCJI OCENY

_ :LOGOUT

_ :FINISH

=< KONIEC PRZEBIEGU MODULU MODYFIKATOR SPR

=====

END OF MACRO

14.27.01_

Po przeprowadzeniu modyfikacji funkcji oceny powtórnie rozwiązano zadanie PLYTKA, wg strategii uporządkowanego wyboru, z nowymi wartościami współczynników λ_i , $i=1,2,\dots,6$ funkcji g. Poniżej zamieszczono tabulogram z przebiegu rozwiązywania. Na uwagę zasługuje fakt, że sekwencja przekształcająca została tym razem osiągnięta szybciej, bo już w sytuacji S72.

10.04.19_ SPR FR

=====

==< SYSTEM POSZUKIWANIA ROZWIĄZAN SPR D:22/09/83 T:10/04/53

=< REALIZATOR SPR

_ :LOGIN DIAL-INT-001

=? HASŁO

_ ZS

_ :FUNCTION G

LAMBDA 1 = 0.504

LAMBDA 2 = 0.500

LAMBDA 3 = 0.973

LAMBDA 4 = 0.905

LAMBDA 5 = 1.000

LAMBDA 6 = 0.818

_ :STRATEGY C

_ :REPORT 0

_ :LIMIT 100

_ :DO PLYTKA

==< PROCEDURA POSZUKIWANIA ROZWIĄZANIA D:22/09/83 T:10/06/31

==< POZIOM RAPORTOWANIA: 0

==< ZADANIE: PLYTKA

=< SYTUACJA POCZĄTKOWA - "S1":

(E1) NARZ(NAWIERT,F3)

(E2) NARZ(WIERTLO,F4)

(E3) NARZ(WIERTLO,F8)

(E4) NARZ(WIERTLO,F10)

(E5) NARZ(WIERTLO,F12)

(E6) NARZ(GWINT,M5)

(E7) GLOWICA(P0)

=< CEL:

(P1) OTWOR(P1,F8,G20)

(P2) OTWOR(P2,F8,G20)

(P3) OTWOR(P3,F10,G20)

(P4) OTWORGW(P4,M5,G10)

(P5) OTWORGW(P5,M5,G10)

(P6) OTWORGW(P6,M5,G10)

(P7) OTWORPC(P7,F8,G20,F12,G5)

(P8) OTWORPC(P8,F8,G20,F12,G5)

=< REGUŁY PRZEKSZTAŁCANIA:

(01) MOCOW

(02) PRZESUW

(03) NAWIERC

(04) WIERC

(05) POGL

(06) GWINTOW

- :GO

=< OSIAGNIETO ROZWIAZANIE W SYTUACJI "S72"

- :RP 2

- :WRITE S72

S72: (E1) NARZ(NAWIERT,F3)
(E2) NARZ(WIERTLO,F4)
(E3) NARZ(WIERTLO,F8)
(E4) NARZ(WIERTLO,F10)
(E5) NARZ(WIERTLO,F12)
(E6) NARZ(GWINT,M5)
(E21) OTWOR(P1,F8,G20)
(E27) OTWOR(P2,F8,G20)
(E33) OTWOR(P3,F10,G20)
(E46) OTWORGW(P4,M5,G10)
(E54) OTWORGW(P5,M5,G10)
(E62) OTWORGW(P6,M5,G10)
(E70) OTWORPC(P7,F8,G20,F12,G5)
(E73) GLOWICA(P8)
(E77) ZAMOC(WIERTLO,F12)
(E78) OTWORPC(P8,F8,G20,F12,G5)

- :LOGOUT

- :FINISH

=< KONIEC PRZEBIEGU MODULU. REALIZATOR SPR

=====

END OF MACRO

10.20.13_

Przykładowo rozwiązane zadanie sygnalizuje jedno z możliwych zastosowań metody - wspomagane lub automatyczne programowanie procesów sekwencyjnych. Mogą to być np. procesy technologiczne, przedstawiane w postaci programów dla obrabiarek sterowanych numerycznie. Możliwe, nowe względem tradycyjnego, podejście do problemu automatycznego programowania obrabiarek sterowanych numerycznie polegałoby na:

- 1^o tworzeniu baz danych z opisami "środowiska obróbkowego", gdzie przekształcenia technologiczne poszczególnych rodzajów przedstawiane są jako reguły przekształcania a opisy narzędzi, oprzyrządowanie, materiały, itp. - jako opis środowiska problemowego;
- 2^o interakcyjnym wspomaganie projektowania procesów technologicznych, w którym projektant formułuje jedynie opis cech

detalu w jego żądanej postaci finalnej (zamiast opisywać każdy krok procesu technologicznego) a warianty rozwiązań generuje komputer.

7. WNIOSKI

Ze sformułowania tezy w rozdziale 1. wynika, że rozprawa doktorska zmierza do wykazania istnienia metody, umożliwiającej automatyczne poszukiwanie sekwencji przekształcającej w przestrzeni stanów i charakteryzującej się użyciem:

- specjalnego języka opisu zadania, pozwalającego na formułowanie zadań dla metody, na dowolnym poziomie ogólności,
 - dwukierunkowego poszukiwania rozwiązania, ograniczającego w istotny sposób ilość rozpatrywanych wariantów przekształceń,
 - funkcji heurystycznych, dla określania perspektywicznych kierunków poszukiwania rozwiązania,
 - mechanizmu zbieżności do rozwiązania quasi-optimalnego,
 - mechanizmu automatycznej modyfikacji funkcji oceny,
- umożliwiającego gromadzenie doświadczeń i uczenie się rozwiązywania analogicznych problemów.

Jak się autorowi rozprawy wydaje, prawdziwość tak sformułowanej tezy doktorskiej została w niniejszej rozprawie wykazana.

Opracowana metoda jest ogólna i może znaleźć zastosowanie w tych dziedzinach, w których występuje możliwość powierzenia komputerowi zadania poszukiwania sekwencji (ciągu) przekształceń. Użyte funkcje ocen nie są zorientowane na konkretne, specjalistyczne zastosowanie, lecz ilustrują możliwość wpływania na proces poszukiwania rozwiązania, poprzez sukcesywne doskonalenie strategii działania w wieloetapowym procesie przystosowywania się metody (uczenia się) do rozwiązywania zadań z danej klasy. Zaproponowany algorytm automatycznej modyfikacji funkcji oceny może znaleźć zastosowanie również w procesie korygowania funkcji ocen, wyrażających się liniową kombinacją funkcji składowych o innych postaciach,

niż użyte w przedstawionej metodzie.

Opracowany język opisu zadania dla metody umożliwia przedstawianie zadań z różnych dziedzin praktycznych, zawierających się w modelu, opisanym w rozdziale 3. W szczególności, często dają się sformułować w sposób, umożliwiający zastosowanie metody, problemy planowania działań robotów, problemy automatycznego i wspomaganego projektowania oraz automatycznego programowania procesów sekwencyjnych, np. procesów technologicznych, w postaci programów dla obrabiarek sterowanych numerycznie. Możliwość tę, w odniesieniu do projektowania procesów technologicznych, obrazuje przykładowe rozwiązywanie zadania w p. 6.6. Sygnalizuje ono możliwe, nowe (względem tradycyjnego) podejście do problemu automatycznego programowania obrabiarek sterowanych numerycznie. Polega ono na:

- tworzeniu baz danych z opisami "środowiska obróbkowego", gdzie przekształcenia technologiczne poszczególnych rodzajów przedstawiane są jako reguły przekształcania a opisy narzędzi, oprzyrządowanie, materiały, itp. - jako opis środowiska problemowego;

- interakcyjnym wspomaganie projektowania procesów technologicznych, w którym projektant formułuje jedynie opis cech detalu w jego żądanej postaci finalnej (zamiast opisywać każdy krok procesu technologicznego) a warianty rozwiązań generuje komputer.

Przedstawione w rozdziale 6. oprogramowanie metody, w postaci systemu poszukiwania rozwiązań SPR, podobnie, jak opisana w rozprawie metoda, nie jest zorientowane na konkretne, techniczne zastosowanie, stanowi ono ogólne narzędzie, pozwalające na stosowanie metody we wspomnianej wyżej klasie zadań. Podstawowym trybem przetwarzania jest konwersacja, co

umożliwia interakcję przy rozwiązywaniu zadań, a tym samym - połączenie szybkości komputera z uniwersalnością metody oraz doświadczeniem użytkownika, którym może być zarówno student w laboratorium informatyki, jak i projektant procesów produkcyjnych w pracowni zakładu przemysłowego.

Na tle przeprowadzonego rozpoznania literaturowego, w szczególności w obrębie dostępnej autorowi literatury amerykańskiej, która na ten temat dostarcza najbardziej miarodajnych informacji - zaprezentowana w rozprawie metoda może zawierać elementy oryginalne. Przekonanie to, niewątpliwie subiektywne, dotyczy szczególnie organizacji poszukiwania w oparciu o funkcje ocen (wariant C metody) oraz organizacji doskonalenia strategii (uczenia się) metody, w oparciu o algorytm automatycznej modyfikacji funkcji oceny.

Autor jest oczywiście przekonany, że istnieje szereg nieznanых mu rozwiązań, wybiegających znacznie dalej, niż to, co stanowi przedmiot niniejszej rozprawy. Rozwiązania te są jednak na ogół objęte tajemnicą, ze względu na ich znaczenie strategiczne lub ekonomiczne.

W ramach niniejszej rozprawy nie rozwiązano oczywiście wszystkich problemów, wiążących się z tezą rozprawy. W szczególności, nie przeprowadzono analizy porównawczej między proponowaną metodą a innymi metodami, znanymi autorowi z literatury. Należy jednak mieć na uwadze, że przeprowadzenie takiej analizy, nawet na gruncie teorii, jest zadaniem niezwykle poważnym. Podstawową trudnością jest brak zobiektywizowanych kryteriów oceny jakości metody. Ponadto, publikowane opisy metod cechuje z reguły duża ogólność i lakoniczność, utrudniająca analizę porównawczą. Można by wprowadzić próbować przeprowadzić porównanie "efektywności" metod na drodze doświadczalnej, lecz sprowadzałoby się to co najmniej do ko-

nieczności oprogramowania wielu metod o różnych organizacjach poszukiwania rozwiązania, co jest zadaniem kosztownym i pracochłonnym nawet wówczas, gdy jest dostępna dobra implementacja języka bardzo wysokiego poziomu.

Przedstawione wyżej powody uniemożliwiają autorowi rozprawy przeprowadzenie analizy porównawczej opracowanej metody z innymi metodami tego typu. W rozprawie ograniczono się więc do rozważań, których wyniki pozwalają jedynie spodziewać się poprawy niektórych parametrów użytkowych systemu komputerowego, wspomagającego poszukiwanie sekwencji przekształcającej według zaproponowanej metody, w stosunku do rozwiązań tradycyjnych.

LITERATURA

- [1] Benerji R.B.: Artificial Intelligence, North - Holland, New York, Oxford 1980.
- [2] Buchanan B.G., Lederberg J.: The Heuristic DENDRAL Program for Explaining Empirical Data. Proceedings of the IFIP Congress 71, Ljubljana, Yugoslavia, 1971.
- [3] Bundy A.(red.): Artificial Intelligence, North - Holland, Amsterdam, New York, Oxford 1978.
- [4] Cieřlik A., Klajn A., Szpunar Z.: System Wspomagania Dydaktyki DIALOS/MLK. Projekt systemu. Raport Centrum Obliczeniowego Politechniki Wrocławskiej 27/79, Wrocław, 1979.
- [5] Cieřlik A., Klajn A., Szpunar Z.: System Wspomagania Dydaktyki DIALOS/INT. Projekt systemu. Raport Centrum Obliczeniowego Politechniki Wrocławskiej 30/80, Wrocław, 1980.
- [6] Davis R., Buchanan B., Shortliffe E.: Production Rules as a Representation for a Knowledge-Based Consultation Program, Stanford AI Laboratory, Memo AIM-266, Computer Science Dept., Stanford University, California, Oct. 1975.
- [7] Derksen J., Rulifson J.F., Waldinger R.J.: The QA4 Language Applied to Robot Planning, Fall Joint Comp. Conf., Proc., s.1181 - 1192, 1972.
- [8] Doran J.: An Approach to Automatic Problem-Solving, Machine Intelligence, 1, Edinburgh, 1967.
- [9] Eastman C.M.: Heuristic Algorithm for Automated Space planning, 2nd Int. Joint Conference on Artificial Intelligence, London, England, September 1971.
- [10] Eastman C.M.: Automated Space Planning, Artificial Intelligence, Vol. 4, 1973.
- [11] Efimow E.I.: Reszatieli intellektualnych zadacz, Izdatielstwo "Nauka", Moskwa, 1982.
- [12] Ernst G.W., Newell A.: GPS: A Case Study in Generality and Problem Solving, Academic Press, 1969.
- [13] Fahlman S.E.: A Planning System for Robot Construction Tasks, Artificial Intelligence, Vol. 5, 1974.
- [14] Feigenbaum E.A.(red.), Feldman J.(red.): Maszyny matematyczne i myřlenie, PWN, Warszawa, 1972.

- [15] Fikes R.E.: Knowledge Representation in Automatic Planning Systems. Perspectives on Computer Science, 10th Anniversary Symposium at the Computer Science Department, Carnegie-Mellon University, 1979, Academic Press, Inc., New York - San Francisco - London, 1977.
- [16] Fikes R.E., Nilsson N.J.: STRIPS: A New Approach to the Application of Theorem Proving in Problem Solving, Artificial Intelligence, Vol. 2, 1971.
- [17] Frydrychiewicz W., ...: Zasady programowania obrabiarek sterowanych numerycznie, Centralny Ośrodek Techniki Obrabiarek Sterowanych Numerycznie, Centralne Biuro Konstrukcyjne Obrabiarek, Warszawa, 1974.
- [18] Gelernter H.: Realization of a Geometry Theorem-Proving Machine, Proceedings of the Int. Conference of Information Processing, UNESCO House, Paris, 1959.
- [19] Gładun W.P.: Ewristyczny poisk w skłónych sredach, Izdatielstwo "Naukowa Dumka", Kiew, 1977.
- [20] Góralski A.(red.): Zadanie metoda rozwiązanie, zbiór 1., WNT, Warszawa, 1977.
- [21] Góralski A.(red.): Zadanie metoda rozwiązanie, zbiór 2., WNT, Warszawa, 1978.
- [22] Góralski A.(red.): Zadanie metoda rozwiązanie, zbiór 3., WNT, Warszawa, 1980.
- [23] Góralski A.(red.): Zadanie metoda rozwiązanie, zbiór 4., WNT, Warszawa, 1982.
- [24] Hewitt C.: Description and Theoretical Analysis (using schemata) of PLANNER: A Language for Proving Theorems and Manipulating Models in a Robot, MIT, AI Laboratory, Cambridge, Massachusetts, 1972.
- [25] Hołyński M.: Sztuczna inteligencja, Wiedza Powszechna, Warszawa, 1979.
- [26] Hunt E.B.: Iskusstwiennyj intellekt, Izdatielstwo "Mir", Moskwa, 1978 (tyt. oryg.: Artificial Intelligence, Academic Press, New York - San Francisco - London, 1975).
- [27] Jeleniewski T., Sielicki A.: Metodologia i komputerowe wspomaganie projektowania technicznego, Politechnika Wrocławska, Wrocław, 1975.

- [28] Jones J.Ch.: Metody projektowania, WNT, Warszawa, 1977.
- [29] Klajn A., Stuchlik R., Szpunar Z.: Aufbau, Arbeitsweise und Nutzung des DIALOS/MLK - Systems im on-line Betrieb, Wissenschaftliche Zeitschrift der Technischen Hochschule Magdeburg, H.3, 1980.
- [30] Klajn A., Szpunar Z.: DIALOS/INT - ein System mit Elementen Künstlicher Intelligenz für die Lösung von Problem - aufgaben, przyjęte do druku w pracach naukowych Technische Hochschule w Magdeburgu, 1982.
- [31] Klajn A., Szpunar Z.: Koncepcja modułu obsługi dialogu dla procedur konwersacyjnych, Prace naukowe Centrum Obliczeniowego Politechniki Wrocławskiej, nr 1, Wrocław, 1980.
- [32] Klajn A., Szpunar Z.: System DIALOS/INT wspomagania dydaktyki z wykorzystaniem zadań problemowych. Informatyka w dydaktyce. Konferencja TNOiK INFOGRYF 80, Kołobrzeg, 1980.
- [33] Klajn A., Szpunar Z.: System wspomagania dydaktyki DIALOS/MLK. Zasady eksploatacji. Raport Centrum Obliczeniowego Politechniki Wrocławskiej 26/79, Wrocław, 1979.
- [34] Klajn A., Szpunar Z.: System wspomagania dydaktyki DIALOS/INT. Zasady eksploatacji. Raport Centrum Obliczeniowego Politechniki Wrocławskiej 31/80, Wrocław, 1980.
- [35] Klajn A., Szpunar Z.: System wspomagania dydaktyki DIALOS/INT. Przewodnik użytkownika. Raport Centrum Obliczeniowego Politechniki Wrocławskiej 33/80, Wrocław, 1980.
- [36] Kowalski R.: A Proof Procedure Using Connection Graphs, Journal of the Association for Computing Machinery, Vol. 22, No. 4, s. 572÷595, October 1975.
- [37] Kowalski R.: Predicate Logic As Programming Language, Information Processing 74, North - Holland Publishing Company, 1974.
- [38] Latombe J.C.: Artificial Intelligence in Computer-Aided Design: The TROPIC System, CAD Systems, s. 61÷120, North - Holland Publishing Company, 1978.
- [39] Libal V.(red.): Organizacja i zarządzanie produkcją, PWE, Warszawa, 1976.

- [40] Manna Z., Waldinger R.: A Deductive Approach to Program Synthesis, ACM Transactions on Programming Languages and Systems, Vol. 2., No. 1, s.90+121, January 1980.
- [41] Manna Z., Waldinger R.: Synthesis: Dreams $\Rightarrow$ Programs, IEEE Transactions on Software Engineering, SE-5, 4, s. 294+328, July 1979.
- [42] Miller F.A.: Improving Heuristic Regression Analysis, 6-th Annual Southeastern Regional Meeting of the ACM, 1967.
- [43] Newell A., Shaw J.C., Simon N.A.: Empirical Explorations of the Logic Theory Machine Heuristic Study, Proceedings of the Western Joint Computer Conference, 1957.
- [44] Newell A., Shaw J., Simon H.: A General Problem-Solving Program For a Computer, Computers and Automation, 1959.
- [45] Newell A., Simon H.A.: Human Problem Solving, Prentice Hall, 1972.
- [46] Nilsson N.J.: A Hierarchical Robot Planning and Execution System, Artificial Intelligence Center, SRI International, Technical Note 76, SRI Project 1187, April 1973.
- [47] Nilsson N.J.: A Production System For Automatic Deduction, Machine Intelligence, 9, Ellis Harwood, Chichester, England, 1979.
- [48] Nilsson N.J.: Problem-Solving Methods in Artificial Intelligence, Mc Graw-Hill Book Company, New York, 1971.
- [49] Popow E.W., Firdman G.R.: Algoritmicheskiye osnovy intellektualnykh robotow i iskusstwiennowo intellekta, Izdatielstwo "Nauka", Moskwa, 1976.
- [50] Reboh R., Sacerdoti E.: A Preliminary QLISP Manual, SRI AI Center, Technical Note 81, Menlo Park, California, 1973.
- [51] Robinson J.A.: A Machine-Oriented Logic Based on the Resolution Principle, Journal of the ACM, Vol. 12., No 1, January 1965.
- [52] Roussel P.: PROLOG. Manuel de reference et d'utilisation. Groupe d'Intelligence Artificielle. UER Marseille-Luminy, Septembre 1975.
- [53] Sacerdoti E.D.: A Structure For Plans and Behavior, Elsevier North - Holland, 1977.

- [54] Sacerdoti E.D.: Planning in a Hierarchy of Abstraction Spaces, Artificial Intelligence, Vol. 5, No 2, 1974.
- [55] Sacerdoti E.D.: Problem Solving Tactics, Artificial Intelligence Center, Technical Note 189, SRI Project 7910, SRI International, July 1979.
- [56] Slagle J.R.: A Heuristic Program that Solves Symbolic Integration Problems in Freshman Calculus, Journal of the ACM, Vol. 10, No 4, 1963.
- [57] Slagle J.R.: Iskusstwiennyj intellekt. Podchod na osnowie evristiczeskowo programmirowania (tyt. oryg.: Artificial Intelligence: The Heuristic Programming Apporach), Izdatielstwo "Mir", Moskwa, 1973.
- [58] Slagle J., Bursky P.: Experiments with a Multipurpose, Theorem-Proving Heuristic Program, Journal of the ACM, Vol. 15, No 1, 1968.
- [59] Sussman G.J., Mc Dermott D.V.: From PLANNER to CONNIVER: A Genetic Approach, Fall Joint Computer Conference, Proc., s. 1171÷1180, 1972.
- [60] Szafarczyk M.: Sterowanie programowe obrabiarek, WNT, Warszawa, 1963.
- [61] Szpunar Z.: Metoda generacji dokumentów w procesie wypro- wadzania informacji z baz danych systemów informatycznych, Prace naukowe Centrum Obliczeniowego Politechniki Wroclaw- skiej, nr 1, 1980.
- [62] Tate A.: INTERPLAN: A Plan Generation System Which Can Deal with Interactions Between Goals, Machine Intelligen- ce Research Unit, University of Edinburgh, December 1974.
- [63] Tymowski J.: Automatyzacja procesów technologicznych w przemyśle maszynowym, WNT, Warszawa, 1975.
- [64] Van Emden M.H., Kowalski R.A.: The Semantics of Predica- te Logic as a Programming Language, Journal of the ACM, Vol. 23, No 4, s. 733÷742, October 1976.
- [65] Waldinger R.: Achieving Several Goals Simultaneously, Machine Intelligence, 8, s. 94÷136, Ellis Horwood Ltd., Chichester, England, 1977.
- [66] Warren D.H.D.: WARPLAN: A System for Generating Plans, Department of Computational Logic, Memo No 76, University of Edinburgh, June 1974.

- [67] Winston P.H.: Iskustwiennyj intellekt (tyt. oryg.: Artificial Intelligence), Izdatielstwo "Mir", Moskwa, 1980.
- [68] Zbierski S.: Technologia montażu maszyn. Projektowanie procesów technologicznych, OBR Podstaw Technologii i Konstrukcji Maszyn, Warszawa, 1980.

