

PRZEDMIOT: Metody Inżynierii Wiedzy – Projekt
TYTUŁ: System regulowy - winda

Jakub Sysak sysak@mops.uci.agh.edu.pl

Paweł Zieliński pziel@biuro.net.pl

AiR, IV rok, lato 2002

16 lipca 2002

Spis treści

1	Opis problemu	3
2	Opis tablicy reguł	4
3	Analiza zbioru reguł	7
3.1	Istotność atrybutów	7
3.2	Duplikaty rekordów	7
3.3	Subsumpcja	7
3.4	Determinizm i niespójność	7
3.5	Redukcja reguł	8
3.6	Zupełność zbioru reguł	11
3.7	Relacja odległości między rekordami	12
3.8	Specyficzne charakterystyki systemu	12
3.9	Charakterystyki i zależności temporalne	12
3.10	Dekompozycja systemu	13
3.11	Mechanizmy adaptacji i uczenia się	13
4	Opis programu	14
4.1	Opis interfejsu	14
4.2	Opis implementacji	15
5	Bibliografia	16

1. Opis problemu

Naszym celem było stworzenie systemu regulowego sterującego zachowaniem windy pasażerskiej. Dla uproszczenia modelu nasza winda nie posiada drzwi otwieranych automatycznie czy też mechanicznie. Wszystkie decyzje odnośnie zachowania windy podejmowane są, gdy winda jest na poziomie któregoś z pięter. Winda reaguje na przyciski wewnątrz kabiny, jak również na przyciski przyzywające ją z pięter. Na każdej kondygnacji oprócz kondygnacji pierwszej i ostatniej znajdują się dwa przyciski, które umożliwiają wybór kierunku jazdy. Na parterze jest tylko przycisk do jazdy w górę, na ostatnim piętrze przycisk do jazdy w dół. Zatrzymanie windy zarówno przy jeździe w dół, jak i do góry odbywa się bez procesu hamowania (w momencie dojazdu na odpowiednie piętro odbywa się natychmiastowe hamowanie). Jest ona wyposażona w czujnik sprawdzający obciążenie windy. Czujnik ten reaguje wysyłając sygnał o przeciążeniu windy w momencie wsiadania pasażerów, gdy zostanie przekroczona granica obciążenia. Winda wówczas nie rusza dopóki część pasażerów nie wysiądzie. Czujnik wykorzystywany jest również w momencie mijania w czasie jazdy pięter, na których winda jest przyzywana. Winda zatrzymuje się na takich piętrach tylko wtedy, gdy jest obciążona poniżej wartości granicznej.

Jednym z podstawowych założeń dotyczących naszego systemu jest to, że sterownik windy podejmuje decyzje co do jej ruchu jedynie na podstawie ogólnej informacji: winda przyzywana z piętra powyżej, poniżej aktualnego poziomu windy lub z aktualnego poziomu. Nie bierze się pod uwagę konkretnego numeru piętra. Z jednej strony sprawia to, że sterownik jest uniwersalny i może działać przy dowolnej ilości pięter. Istnieje jednak też minus takiego podejścia. Brak informacji o numerze piętra, na które ma jechać winda nie pozwala obliczyć jaką drogę ma ona do przebycia. Wiąże się to z niemożliwością zastosowania algorytmu, który przy jednoczesnym wezwaniu windy do jazdy w dwóch kierunkach oceniałby, które wezwanie jest wstanie obsłużyć szybciej i na tej podstawie wybierałby kierunek jazdy.

Stworzony przez nas algorytm działania windy opiera się na zasadzie, że winda podążą w jednym kierunku tak długo, jak długo istnieją wezwania związane z tym kierunkiem. Dopiero później winda przystępuje do obsługi zadań związanych z przeciwnym do aktualnego kierunkiem jazdy. Od tej reguły istnieje jeden wyjątek. Winda zmienia kierunek jazdy pomimo wezwań z pięter, gdy w kabinie windy zostanie wciśnięty przycisk żądający jazdy w przeciwną stronę. Pasażerowie znajdujący się w windzie mają pierwszeństwo nad osobami przyzywającymi windę z pięter. Zastosowaliśmy tą zasadę aby uniknąć sytuacji, w której winda będzie jeździć z pełnym obciążeniem nie realizując żądań z kabiny i jednocześnie uniemożliwiając wejście do środka ludziom oczekującym na piętrach.

2. Opis tablicy reguł

W naszym systemie reguł występuje 14 atrybutów: 12 atrybutów prewarunków i 2 atrybuty konkluzji. Wśród atrybutów prewarunków można wyróżnić 9 atrybutów binarnych i 3 atrybuty o trzelementowym zbiorze wartości. Atrybuty binarne odpowiadają za określenie stanu przycisków przyzywających windę. Przyjmują one wartości J lub N, co odpowiada temu, że przycisk jest lub nie jest wciśnięty. Jeden z atrybutów o trzelementowym zbiorze wartości określa obecny stan windy, drugi poprzedni stan windy. Mogą one przyjmować te same wartości S - postój, JG - jazda do góry i JD - jazda na dół. Ostatni z atrybutów prewarunków określa obciążenie windy. Przyjmuje on wartości: D - obciążenie dopuszczalne, G - obciążenie graniczne, P - obciążenie przekroczone. Z dwóch atrybutów konkluzji jeden jest binarny, a drugi posiada trzelementowy zbiór wartości. Atrybut binarny odpowiada za włączanie alarmu informującego o przeciążeniu windy. Przyjmuje on wartości J - alarm włączony, N - alarm wyłączony. Drugi z atrybutów konkluzji określa jaką akcję ma podjąć winda: S - zatrzymać się, JG - jechać do góry, czy JD - jechać w dół.

Atrybuty prewarunków:

- W1 - obecny ruch windy: stoi, jedzie do góry, jedzie na dół (S, JG, JD)
- W2 - poprzedni ruch windy: stoi, jedzie do góry, jedzie na dół (S, JG, JD)
- W3 - piętro powyżej windy wybrane z kabiny: jest, nie ma (J, N)
- W4 - piętro poniżej windy wybrane z kabiny: jest, nie ma (J, N)
- W5 - piętro na poziomie windy wybrane z kabiny: jest, nie ma (J, N)
- W6 - wezwanie windy z piętra powyżej do jazdy do góry: jest, nie ma (J, N)
- W7 - wezwanie windy z piętra powyżej do jazdy na dół: jest, nie ma (J, N)
- W8 - wezwanie windy z piętra poniżej do jazdy do góry: jest, nie ma (J, N)
- W9 - wezwanie windy z piętra poniżej do jazdy na dół: jest, nie ma (J, N)
- W10 - wezwanie windy z piętra na poziomie windy do jazdy do góry: jest, nie ma (J, N)
- W11 - wezwanie windy z piętra na poziomie windy do jazdy na dół: jest, nie ma (J, N)
- W12 - obciążenie windy: dopuszczalne, graniczne, przekroczone (D, G, P)

Atrybuty konkluzji:

- A1 - ruch windy: winda stoi, jedzie do góry, jedzie na dół (S, JG, JD)
- A2 - sygnał o przeciążeniu windy: jest, nie ma (J, N)

Każdy z atrybutów posiada dziedzinę składającą się wyłącznie z elementów atomicznych. W związku z tym w naszym systemie tablicowym nie występują żadne interesujące zależności pomiędzy elementami dziedziny.

Pierwotna tablica zawiera 47 reguł:

Nr	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	A1	A2
1	S	S	N	N	-	N	N	N	N	-	-	-	S	N
2	S	S	J	-	-	-	-	-	-	-	-	P	S	J
3	S	S	J	-	-	-	-	-	-	-	-	G	JG	N
4	S	S	J	-	-	-	-	-	-	-	-	D	JG	N
5	S	S	N	J	-	-	-	-	-	-	-	P	S	J
6	S	S	N	J	-	-	-	-	-	-	-	G	JD	N
7	S	S	N	J	-	-	-	-	-	-	-	D	JD	N
8	S	S	N	N	-	J	-	-	-	-	-	-	JG	N
9	S	S	N	N	-	-	J	-	-	-	-	-	JG	N
10	S	S	N	N	-	N	N	J	-	-	-	-	JD	N
11	S	S	N	N	-	N	N	-	J	-	-	-	JD	N
12	S	JG	J	-	-	-	-	-	-	-	-	P	S	J
13	S	JG	J	-	-	-	-	-	-	-	-	G	JG	N
14	S	JG	J	-	-	-	-	-	-	-	-	D	JG	N
15	S	JG	N	J	-	-	-	-	-	-	-	P	S	J
16	S	JG	N	J	-	-	-	-	-	-	-	G	JD	N
17	S	JG	N	J	-	-	-	-	-	-	-	D	JD	N
18	S	JG	N	N	-	J	-	-	-	-	-	-	JG	N
19	S	JG	N	N	-	-	J	-	-	-	-	-	JG	N
20	S	JG	N	N	-	N	N	J	-	-	-	-	JD	N
21	S	JG	N	N	-	N	N	-	J	-	-	-	JD	N
22	S	JD	-	J	-	-	-	-	-	-	-	P	S	J
23	S	JD	-	J	-	-	-	-	-	-	-	G	JD	N
24	S	JD	-	J	-	-	-	-	-	-	-	D	JD	N
25	S	JD	J	N	-	-	-	-	-	-	-	P	S	J
26	S	JD	J	N	-	-	-	-	-	-	-	G	JG	N
27	S	JD	J	N	-	-	-	-	-	-	-	D	JG	N
28	S	JD	N	N	-	-	-	-	J	-	-	-	JD	N
29	S	JD	N	N	-	-	-	J	-	-	-	-	JD	N
30	S	JD	N	N	-	J	-	N	N	-	-	-	JG	N
31	S	JD	N	N	-	-	J	N	N	-	-	-	JG	N
32	JG	-	-	-	J	-	-	-	-	-	-	-	S	N
33	JG	-	J	-	-	-	-	-	-	J	-	G	JG	N
34	JG	-	J	-	-	-	-	-	-	J	-	D	S	N
35	JG	-	N	-	-	-	-	-	-	J	-	-	S	N
36	JG	-	N	-	-	N	N	-	-	-	J	-	S	N
37	JG	-	J	-	N	-	-	-	-	N	-	-	JG	N
38	JG	-	-	-	N	J	-	-	-	N	-	-	JG	N
39	JG	-	-	-	N	-	J	-	-	N	-	-	JG	N
40	JD	-	-	-	J	-	-	-	-	-	-	-	S	N
41	JD	-	-	J	-	-	-	-	-	-	J	G	JD	N

Nr	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	A1	A2
42	JD	-	-	J	-	-	-	-	-	-	J	D	S	N
43	JD	-	-	N	-	-	-	-	-	-	J	-	S	N
44	JD	-	-	N	-	-	-	N	N	J	-	-	S	N
45	JD	-	-	J	N	-	-	-	-	-	N	-	JD	N
46	JD	-	-	-	N	-	-	-	J	-	N	-	JD	N
47	JD	-	-	-	N	-	-	J	-	-	N	-	JD	N

Reguły zawarte w tablicy nie pokrywają wszystkich możliwych stanów powstałych w wyniku iloczynu kartezjańskiego dziedzin atrybutów. Istnieją takie kombinacje wartości atrybutów, które w rzeczywistości nigdy się nie pojawią. Zawiera je poniższa tablica:

W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12
JG	-	N	-	N	N	N	-	-	N	N	-
JD	-	-	N	N	-	-	N	N	N	N	-
JG	-	J	-	N	-	-	-	-	J	-	P
JD	-	-	J	N	-	-	-	-	-	J	P

Ilość elementów iloczynu kartezjańskiego występujących w opisywanej tablicy decyzyjnej atrybutów prewarunków wynosi 13824. Tablica stanów niedopuszczalnych pokrywa 528 stanów. W związku z tym, jeżeli system ma być zupełny tablica decyzyjna powinna pokrywać 13296 stanów. Gęstość bazy wynosi zatem 0.96.

Opisywana i badana przez nas tablica reguł jest naszego autorstwa. Została ona opracowana na podstawie zdroworozsądkowej analizy możliwych do zaistnienia stanów, w jakich winda może się znaleźć i przydzielenia im odpowiednich decyzji, co do zachowania windy. Jako autorzy tej tablicy udzielamy zgody na dowolne jej wykorzystanie informując jednocześnie, że nie bierzemy żadnej odpowiedzialności za następstwa powstałe w wyniku jej użycia.

3. Analiza zbioru reguł

Przy analizowaniu zbioru reguł korzystaliśmy z informacji z wykładu z Inżynierii Wiedzy[1]. Pomocnym okazało się też opracowanie dotyczące analizy zbioru reguł znalezione na stronach projektu Regulus[2].

3.1. Istotność atrybutów

Nie udało nam się znaleźć atrybutów, które byłyby zbędne. By system regułowy posiadał zadaną funkcjonalność, do podejmowania decyzji potrzeba informacji o stanie wszystkich atrybutów występujących w tablicy.

3.2. Duplikaty rekordów

W stworzonej przez nas tablicy nie występują powtarzające się wiersze.

3.3. Subsumpcja

Ponieważ dziedziny wszystkich atrybutów w tablicy zawierają elementy atomiczne, nie zachodzi zjawisko subsumpcji.

3.4. Determinizm i niespójność

W czasie analizy reguł wykryliśmy dwie pary reguł, które przy tych samych wartościach atrybutów prewarunków podejmują inne decyzje. Są to reguły 32 i 33 oraz 40 i 41:

Nr	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	A1	A2
32	JG	-	-	-	J	-	-	-	-	-	-	-	S	N
33	JG	-	J	-	-	-	-	-	-	J	-	G	JG	N

Nr	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	A1	A2
40	JD	-	-	-	J	-	-	-	-	-	-	-	S	N
41	JD	-	-	J	-	-	-	-	-	-	J	G	JD	N

Aby zbiór reguł był spójny i deterministyczny wprowadziliśmy zmiany do reguł 33 i 41. Wskazana wydała nam się też zmiana reguł 34 i 42 ze względu na logikę podejmowania decyzji. Jeśli dla danego piętra jest żądanie zatrzymania windy z kabiny, winda się zatrzymuje. Inne warunki są sprawdzane tylko wtedy, gdy nie ma żądania zatrzymania windy ze środka kabiny. Zmodyfikowane reguły wyglądają w następujący sposób:

Nr	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	A1	A2
33	JG	-	J	-	N	-	-	-	-	J	-	G	JG	N
34	JG	-	J	-	N	-	-	-	-	J	-	D	S	N

Nr	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	A1	A2
41	JD	-	-	J	N	-	-	-	-	-	J	G	JD	N
42	JD	-	-	J	N	-	-	-	-	-	J	D	S	N

3.5. Redukcja reguł

Ponieważ większość atrybutów prewarunków jest binarna, zdecydowaliśmy się na zastosowanie redukcji bez sklejania: nie dało się sklejać reguł tworząc iloczyn wartości atrybutów. Cały zbiór reguł podzieliliśmy na podzbiory ze względu na podejmowane decyzje, po czym przystąpiliśmy do sklejania.

Dla $A1=S$ i $A2=N$ sklejeniu uległy reguły 32 i 40:

Nr	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	A1	A2
32	JG	-	-	-	J	-	-	-	-	-	-	-	S	N
40	JD	-	-	-	J	-	-	-	-	-	-	-	S	N
	JGvJD	-	-	-	J	-	-	-	-	-	-	-	S	N

Dla $A1=S$ i $A2=J$ sklejeniu uległy reguły 2 i 12 oraz 5 i 15:

Nr	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	A1	A2
2	S	S	J	-	-	-	-	-	-	-	-	P	S	J
12	S	JG	J	-	-	-	-	-	-	-	-	P	S	J
	S	SvJG	J	-	-	-	-	-	-	-	-	P	S	J

Nr	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	A1	A2
5	S	S	N	J	-	-	-	-	-	-	-	P	S	J
15	S	JG	N	J	-	-	-	-	-	-	-	P	S	J
	S	SvJG	N	J	-	-	-	-	-	-	-	P	S	J

Dla $A1=JG$ i $A2=N$ sklejeniu uległy reguły 3 i 4, 8 i 18, 9 i 19, 13 i 14 oraz 26 i 27:

Nr	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	A1	A2
3	S	S	J	-	-	-	-	-	-	-	-	G	JG	N
4	S	S	J	-	-	-	-	-	-	-	-	D	JG	N
	S	S	J	-	-	-	-	-	-	-	-	GvD	JG	N

Nr	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	A1	A2
8	S	S	N	N	-	J	-	-	-	-	-	-	JG	N
18	S	JG	N	N	-	J	-	-	-	-	-	-	JG	N
	S	SvJG	N	N	-	J	-	-	-	-	-	-	JG	N

Nr	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	A1	A2
9	S	S	N	N	-	-	J	-	-	-	-	-	JG	N
19	S	JG	N	N	-	-	J	-	-	-	-	-	JG	N
	S	SvJG	N	N	-	-	J	-	-	-	-	-	JG	N

Nr	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	A1	A2
13	S	JG	J	-	-	-	-	-	-	-	-	G	JG	N
14	S	JG	J	-	-	-	-	-	-	-	-	D	JG	N
	S	JG	J	-	-	-	-	-	-	-	-	GvD	JG	N

Nr	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	A1	A2
26	S	JD	J	N	-	-	-	-	-	-	-	G	JG	N
27	S	JD	J	N	-	-	-	-	-	-	-	D	JG	N
	S	JD	J	N	-	-	-	-	-	-	-	GvD	JG	N

Dla A1=JD i A2=N sklejaniu uległy reguły 6 i 7, 10 i 20, 11 i 21, 16 i 17 oraz 23 i 24:

Nr	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	A1	A2
6	S	S	N	J	-	-	-	-	-	-	-	G	JD	N
7	S	S	N	J	-	-	-	-	-	-	-	D	JD	N
	S	S	N	J	-	-	-	-	-	-	-	GvD	JD	N

Nr	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	A1	A2
10	S	S	N	N	-	N	N	J	-	-	-	-	JD	N
20	S	JG	N	N	-	N	N	J	-	-	-	-	JD	N
	S	SvJG	N	N	-	N	N	J	-	-	-	-	JD	N

Nr	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	A1	A2
11	S	S	N	N	-	N	N	-	J	-	-	-	JD	N
21	S	JG	N	N	-	N	N	-	J	-	-	-	JD	N
	S	SvJG	N	N	-	N	N	-	J	-	-	-	JD	N

Nr	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	A1	A2
16	S	JG	N	J	-	-	-	-	-	-	-	G	JD	N
17	S	JG	N	J	-	-	-	-	-	-	-	D	JD	N
	S	JG	N	J	-	-	-	-	-	-	-	GvD	JD	N

Nr	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	A1	A2
23	S	JD	-	J	-	-	-	-	-	-	-	G	JD	N
24	S	JD	-	J	-	-	-	-	-	-	-	D	JD	N
	S	JD	-	J	-	-	-	-	-	-	-	GvD	JD	N

Tablicę udało się zredukować o 13 reguł.

Nr	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	A1	A2
1	S	S	N	N	-	N	N	N	N	-	-	-	S	N
2	S	JD	-	J	-	-	-	-	-	-	-	P	S	J
3	S	JD	J	N	-	-	-	-	-	-	-	P	S	J
4	S	JD	N	N	-	-	-	-	J	-	-	-	JD	N
5	S	JD	N	N	-	-	-	J	-	-	-	-	JD	N
6	S	JD	N	N	-	J	-	N	N	-	-	-	JG	N
7	S	JD	N	N	-	-	J	N	N	-	-	-	JG	N
8	JG	-	J	-	N	-	-	-	-	J	-	G	JG	N
9	JG	-	J	-	N	-	-	-	-	J	-	D	S	N
10	JG	-	N	-	-	-	-	-	-	J	-	-	S	N
11	JG	-	N	-	-	N	N	-	-	-	J	-	S	N
12	JG	-	J	-	N	-	-	-	-	N	-	-	JG	N
13	JG	-	-	-	N	J	-	-	-	N	-	-	JG	N
14	JG	-	-	-	N	-	J	-	-	N	-	-	JG	N
15	JD	-	-	J	N	-	-	-	-	-	J	G	JD	N
16	JD	-	-	J	N	-	-	-	-	-	J	D	S	N
17	JD	-	-	N	-	-	-	-	-	-	J	-	S	N
18	JD	-	-	N	-	-	-	N	N	J	-	-	S	N
19	JD	-	-	J	N	-	-	-	-	-	N	-	JD	N
20	JD	-	-	-	N	-	-	-	J	-	N	-	JD	N
21	JD	-	-	-	N	-	-	J	-	-	N	-	JD	N
22	JGvJD	-	-	-	J	-	-	-	-	-	-	-	S	N
23	S	SvJG	J	-	-	-	-	-	-	-	-	P	S	J
24	S	SvJG	N	J	-	-	-	-	-	-	-	P	S	J
25	S	S	J	-	-	-	-	-	-	-	-	GvD	JG	N
26	S	JG	J	-	-	-	-	-	-	-	-	GvD	JG	N
27	S	JD	J	N	-	-	-	-	-	-	-	GvD	JG	N
28	S	SvJG	N	N	-	J	-	-	-	-	-	-	JG	N
29	S	SvJG	N	N	-	-	J	-	-	-	-	-	JG	N
30	S	S	N	J	-	-	-	-	-	-	-	GvD	JD	N
31	S	SvJG	N	N	-	N	N	J	-	-	-	-	JD	N
32	S	SvJG	N	N	-	N	N	-	J	-	-	-	JD	N
33	S	JG	N	J	-	-	-	-	-	-	-	GvD	JD	N
34	S	JD	-	J	-	-	-	-	-	-	-	GvD	JD	N

3.6. Zupełność zbioru reguł

Reguły zawarte w badanej tablicy nie pokrywają całej przestrzeni potencjalnie możliwych stanów wejściowych. Istnieją stany nie obsługiwane przez system reguł z powodu niemożliwości ich wystąpienia w rzeczywistości. W związku z tym zbadaliśmy zupełność specyficzną zbioru reguł.

W systemie decyzje podejmowane są na podstawie wartości 9 atrybutów binarnych i 3 atrybutów, których zbiory wartości są trzejelementowe. Daje to w sumie 13824 stany. 528 stany, to stany niedopuszczalne, których zbiór reguł nie obsługuje. W związku z tym reguły powinny pokrywać 13296 stanów.

W czasie analizy zupełności wykryliśmy, że brakuje reguł dla sytuacji, w której winda stoi i nie ma żadnych wezwań na pietra powyżej lub poniżej niej, a chwilę wcześniej była w ruchu w związku z tym zmodyfikowaliśmy regułę numer 1. Ostateczna tablica reguł ma postać:

Nr	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	A1	A2
1	S	-	N	N	-	N	N	N	N	-	-	-	S	N
2	S	JD	-	J	-	-	-	-	-	-	-	P	S	J
3	S	JD	J	N	-	-	-	-	-	-	-	P	S	J
4	S	JD	N	N	-	-	-	-	J	-	-	-	JD	N
5	S	JD	N	N	-	-	-	J	-	-	-	-	JD	N
6	S	JD	N	N	-	J	-	N	N	-	-	-	JG	N
7	S	JD	N	N	-	-	J	N	N	-	-	-	JG	N
8	JG	-	J	-	N	-	-	-	-	J	-	G	JG	N
9	JG	-	J	-	N	-	-	-	-	J	-	D	S	N
10	JG	-	N	-	-	-	-	-	-	J	-	-	S	N
11	JG	-	N	-	-	N	N	-	-	-	J	-	S	N
12	JG	-	J	-	N	-	-	-	-	N	-	-	JG	N
13	JG	-	-	-	N	J	-	-	-	N	-	-	JG	N
14	JG	-	-	-	N	-	J	-	-	N	-	-	JG	N
15	JD	-	-	J	N	-	-	-	-	-	J	G	JD	N
16	JD	-	-	J	N	-	-	-	-	-	J	D	S	N
17	JD	-	-	N	-	-	-	-	-	-	J	-	S	N
18	JD	-	-	N	-	-	-	N	N	J	-	-	S	N
19	JD	-	-	J	N	-	-	-	-	-	N	-	JD	N
20	JD	-	-	-	N	-	-	-	J	-	N	-	JD	N
21	JD	-	-	-	N	-	-	J	-	-	N	-	JD	N
22	JGvJD	-	-	-	J	-	-	-	-	-	-	-	S	N
23	S	SvJG	J	-	-	-	-	-	-	-	-	P	S	J
24	S	SvJG	N	J	-	-	-	-	-	-	-	P	S	J
25	S	S	J	-	-	-	-	-	-	-	-	GvD	JG	N
26	S	JG	J	-	-	-	-	-	-	-	-	GvD	JG	N
27	S	JD	J	N	-	-	-	-	-	-	-	GvD	JG	N
28	S	SvJG	N	N	-	J	-	-	-	-	-	-	JG	N
29	S	SvJG	N	N	-	-	J	-	-	-	-	-	JG	N
30	S	S	N	J	-	-	-	-	-	-	-	GvD	JD	N

Nr	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	A1	A2
31	S	SvJG	N	N	-	N	N	J	-	-	-	-	JD	N
32	S	SvJG	N	N	-	N	N	-	J	-	-	-	JD	N
33	S	JG	N	J	-	-	-	-	-	-	-	GvD	JD	N
34	S	JD	-	J	-	-	-	-	-	-	-	GvD	JD	N

Ilości stanów pokrywane przez poszczególne reguły:

1. 72
2. 256
3. 72
3. 128
4. i 5. 288
6. i 7. 72
8. 192
9. 192
10. i 11 odliczając stany pokrywane przez 22. 648
12. ,13. i 14. 1008
15. 192
16. 192
17. i 18 odliczając stany pokrywane przez 22. 648
19. , 20. i 21. 1008
22. 4608
23. 512
24. 256
25. 512
26. 512
27. 256
28. i 29. 576
30. 256
31. i 32. 144
33. 256
34. 512

W sumie daje to 13296. Reguły pokrywają więc wszystkie stany dopuszczalne.

3.7. Relacja odległości między rekordami

W przypadku badanej tablicy określanie relacji odległości pomiędzy rekordami wydaje się trudne i nie mające większego sensu.

3.8. Specyficzne charakterystyki systemu

System reguł jest w pełni deterministyczny. Żadne decyzje nie są podejmowane w oparciu o teorię prawdopodobieństwa, niepewności, czy zbiorów rozmytych.

3.9. Charakterystyki i zależności temporalne

Działanie systemu nie jest zależne od czasu.

3.10. Dekompozycja systemu

Atrybuty prewarunków i konkluzji są ze sobą powiązane w ten sposób, że nie ma możliwości zdekomponowania systemu do kilku tabel o prostszej budowie.

3.11. Mechanizmy adaptacji i uczenia się

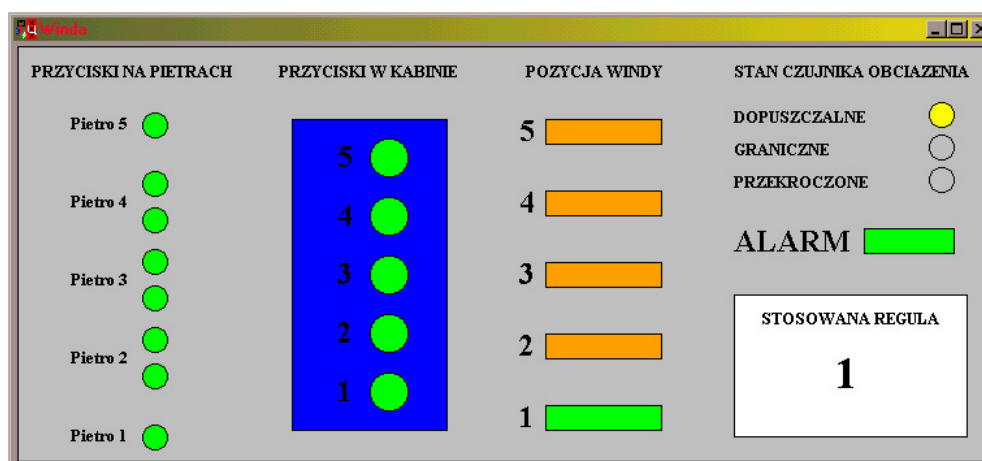
Nasz system decyzyjny nie przewiduje możliwości wprowadzenia mechanizmów adaptacji i uczenia się. Od sterownika windy nie wymaga się, aby jego algorytm posiadał możliwość adaptacji lub uczenia się. Musi on obsługiwać wszystkie możliwe stany i działać bezbłędnie od samego początku pracy.

4. Opis programu

Kolejnym etapem projektu było stworzenie programu w języku Prolog, który pozwoliłby przetestować zachowanie naszego systemu regulowego. Wykorzystaliśmy do tego celu darmowy kompilator SWI-Prolog w wersji 5.0.5 wraz z dołączoną do niego biblioteką graficzną XPCE. Przy tworzeniu programu korzystaliśmy z podręcznika do SWI-Prolog 5.0 [3] oraz z podręcznika programowania w XPCE [4]. Napisany przez nas program umożliwia sprawdzenie działania reguł dla przypadku windy obsługującej 5 kondygnacji. Program uruchamia się przez wywołanie predykatu *start*.

4.1. Opis interfejsu

Po uruchomieniu programu pojawia się okno dialogowe, na którym znajdują się przyciski przyzywające windę na poszczególne piętra z kabiny, jak i z pięter. Pojawiają się na nim także przyciski umożliwiające zmianę obciążenia windy. Okno dialogowe umożliwia śledzenie aktualnego położenia windy, a także wypisuje informację o numerach kolejnych reguł, które są wykorzystywane do sterowania windą. Przyciski w kabinie windy, jak i na piętrach po ich na-



Rysunek 1: Okno dialogowe programu Winda.

ciśnięciu zmieniają kolor na czerwony. Oznacza to, że wezwanie windy zostało przyjęte przez program. Gdy winda zatrzyma się na piętrze, na które była wzywana, przyciski zmieniają swój kolor z powrotem na zielony. Aktualne położenie windy jest pokazywane przy pomocy pięciu prostokątów. Gdy winda znajduje się na piętrze reprezentowanym przez dany prostokąt może on przybrać kolor zielony lub czerwony. Zielony, gdy winda stoi i czerwony, gdy winda tylko mija dane piętro. Reszta prostokątów w tym czasie ma kolor pomarańczowy. Przycisk z aktualnym obciążeniem windy ma kolor żółty, pozostałe przyciski są szare. Gdy system regulowy podejmie decyzję o włączeniu alarmu informującego o przeciążeniu windy prostokąt ALARM zmienia kolor na czerwony.

4.2. Opis implementacji

Predykat uruchamiający program to predykat *start*. Tworzy on okno dialogowe i umieszcza na nim wszystkie elementy graficzne. Uruchamia timer *@timer* oraz wywołuje predykat *inicjalizuj*. Wszystkie elementy graficzne, które zmieniają w czasie pracy programu swoje parametry (kolor, treść napisu) są elementami globalnymi, utworzonymi przy pomocy predykatu *pce_global*. Dzięki temu mają do nich dostęp wszystkie predykaty naszego programu.

Wszelkie informacje o stanie windy i przycisków są przechowywane w bazie danych. Predykat *inicjalizuj* wywołuje predykat *czyscbaze*, który usuwa wszystkie informacje z bazy, a następnie umieszcza w bazie informacje odpowiadające stanowi początkowemu windy.

Przy włączeniu programu uruchamiany jest timer *@timer*, który co trzy sekundy uruchamia predykat *jazda*. Predykat *jazda* odpowiada za pracę windy. Pobiera on z bazy danych informacje o stanie, w którym znalazła się winda oraz które przyciski są wciśnięte, a następnie wykorzystując reguły (wywołanie predykatu *rule*) podejmuje decyzję co do zachowania windy. Ponieważ w systemie regułowym decyzje podejmowane są na podstawie informacji, które mówią tylko o przyciskach powyżej lub poniżej windy, nie precyzując numerów pięter, potrzebne są predykaty, które będą informację przechowywaną w bazie danych tłumaczyć na informację, której potrzebuje system regułowy. Tę rolę pełnią predykaty *w3*, *w4*, *w5*, *w6*, *w7*, *w8*, *w9*, *w10* i *w11*. Każdy z nich oblicza wartość kolejnego atrybutu: odpowiednio *w3* atrybutu *W3*, *w4* atrybutu *W4* itd. Wynik zwracany przez predykat *rule* jest wykorzystywany przez predykaty *nowepietro*, który oblicza nowe położenie windy, *alarm*, który obsługuje znacznik alarmu w oknie dialogowym, oraz predykat *kolory*, którego zadaniem jest obsługa przycisków i znaczników pięter. Informacje o nowym stanie windy są umieszczane w bazie danych.

Tablica reguł jest przeniesiona do programu przy pomocy predykatu *rule*. Ma on dokładnie takie atrybuty, jak kolejne reguły. Został dodany jedynie jeden parametr, który określa numer reguły, którą reprezentuje dany predykat. Tak na przykład wygląda predykat realizujący regułę numer jeden w tablicy reguł:

```
rule(s,_,n,n,_,n,n,n,n,_,_,_,s,n,1):- !.
```

Predykat *kolory* wywołuje dwa następne predykaty: *kolorypieter* i *koloryprzyciskow*. Predykat *kolorypieter* na podstawie informacji na którym piętrze winda się znajduje oraz czy stoi zmienia kolory odpowiednich znaczników pięter. Predykat *koloryprzyciskow* robi to samo z przyciskami na piętrach oraz w kabinie. Zmienia on też informację na temat stanu przycisków w bazie danych.

5. Bibliografia

Literatura

- [1] prof. dr hab. inż. Antoni Ligęza, Wykład z przedmiotu Inżynieria Wiedzy
- [2] Tomasz Potempa i Łukasz Rokita, Projekt o temacie Zupełność z przedmiotu Inżynieria Wiedzy <http://spock.ia.agh.edu.pl/grant/gjn-Regulus-WWW/miw2001/MiW01-projekty/MiW01-Zupelnosc/MiW01-Zupelnosc.html>
- [3] Podrecznik do implementacji języka Prolog SWI-Prolog 5.0 <http://www.swi.psy.uva.nl/cgi-bin//nph-download/SWI-Prolog/refman/refman.pdf>
- [4] Podrecznik do programowania w XPCE <http://www.swi.psy.uva.nl/cgi-bin//nph-download/xpce/doc/userguide/userguide.pdf>