

## ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ  
УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ



Кафедра ЕОМ  
Кваліфікаційна робота  
Другий (магістерський) рівень

## Вбудована система комбінованого нечіткого виведення в завданнях керування роботами

Автор: Керівник  
Ст. гр. СПм-22-6 Пров. каф. ЕОМ  
Олійник Д.Г. Каргін А.О.

ХНУРЕ, Кафедра ЕОМ  
ХАРКІВ 20 24

1

## Мета кваліфікаційної роботи

Розробити алгоритм комбінованого  
зворотного виводу

- алгоритм повинен приймати у якості вхідних даних: БЗ у форматі правил «ЯКЩО-ТО», БД із відомими фактами, кінцевий факт як ціль пошуку;
- алгоритм має відстежувати знайдені проміжні факти для зменшення кількості обчислень;
- алгоритм має уточнювати ФВ шляхів у процесі їх доповнення знайденими необхідними фактами;
- алгоритм повинен мати можливість відновлення шляху, якщо вже були знайдені відповідні правила;
- результатом роботи алгоритму має бути множина шляхів для розрахунку цільового факту із розрахованими ФВ кожного шляху та із КФВ.

Описати апаратну та програмну  
модель прототипу системи

- описати компоненти апаратної частини моделі прототипу системи;
- описати архітектуру апаратної частини моделі прототипу системи;
- описати архітектуру та компоненти програмної частини прототипу системи;
- описати алгоритм неперервного планування;
- описати шаблон програмного управління АІБС.

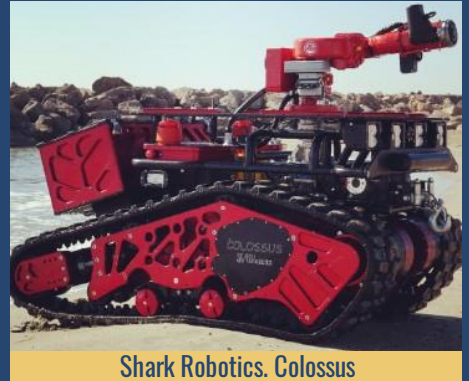
2



Boston Dynamics. Atlas



FANUC. M-1000iA



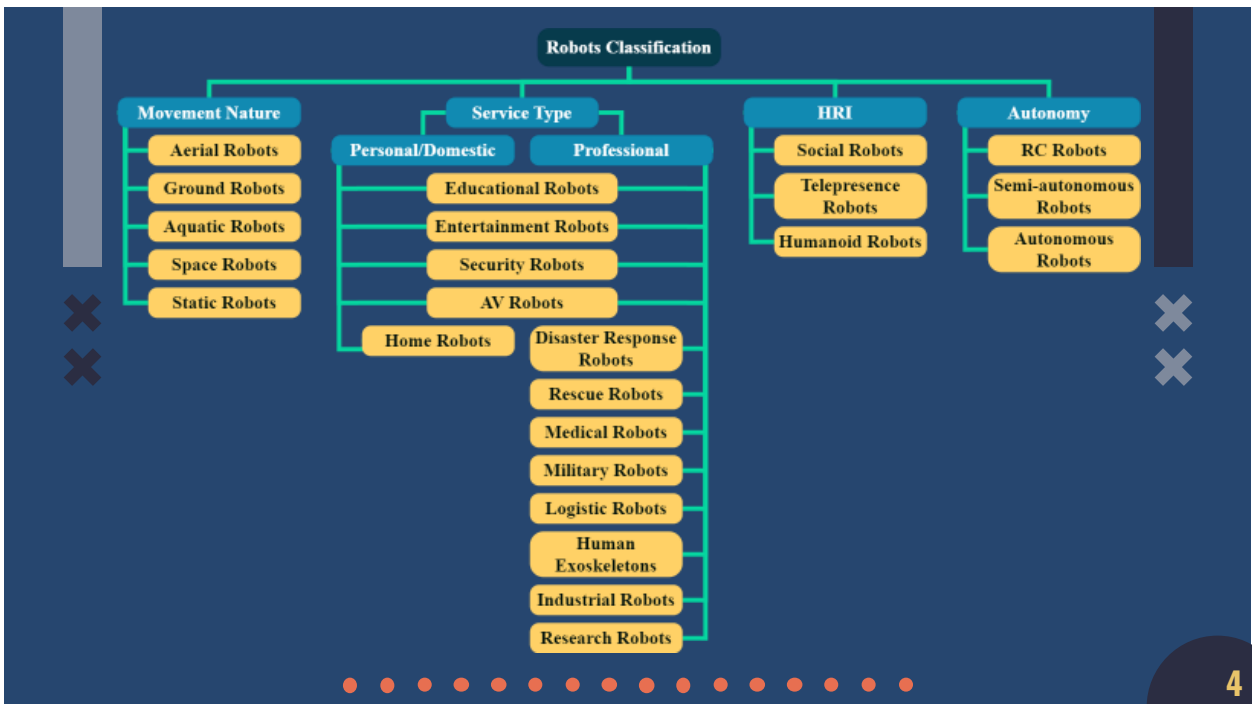
Shark Robotics. Colossus



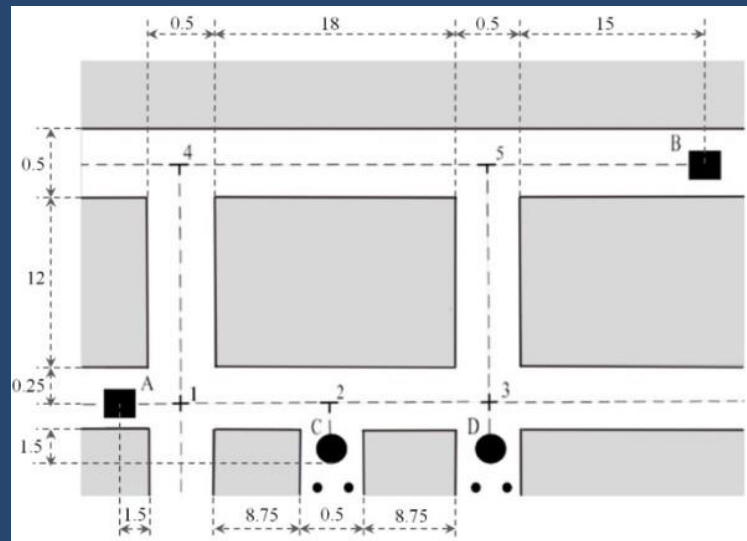
Boston Dynamics. BigDog



FANUC. M-3iA-6A



→ A Load → 1 Move → 4 Move → 5 Move → B Unload → 5 Move → 4 Move → 1 Move → A Load → (1.1)



5

$\langle N, R_N, P_N, \{c_i: \forall c_i \in C_N\}, D_N, CF_N, \exists CF_B \leftrightarrow C_N \in \emptyset \rangle$

(2.1)

де  $N$  – ідентифікатором вузла;

$R_N$  – ідентифікатором правила, за яким розраховується вузол  $N$ ;

$P_N$  – батьківським вузлом  $N$ ;

$C_N$  – множиною дочірніх вузлів  $N$ ;

$D_N$  – множиною залежностей, які необхідно мати для розрахунку вузла  $N$  за правилом  $R_N$ ;

$CF_N$  – поточним розрахованим ФВ вузла  $N$ ;

$CF_B$  – поточним розрахованим ФВ шляху (гілки дерева), до якої входить вузол  $N$  за умови, що вузол  $N$  є листям дерева.

6



$$\{ \langle f_j, \{T_{fk}\} \rangle, \forall f_j \in F_{RN} \} \quad (2.2)$$

де  $\{T_{fk}\}$  – множиною кортежів зі структурою (2.3);

$F_{RN}$  – множина необхідних фактів, що наявні у правилі  $R_N$  вузла  $N$ ;

$$\langle r, CF \rangle, r \in KB_f \quad (2.3)$$

де  $CF$  – поточний ФВ правила  $r$ ;

$KB_f$  – множина правил для визначення факту  $f$ .

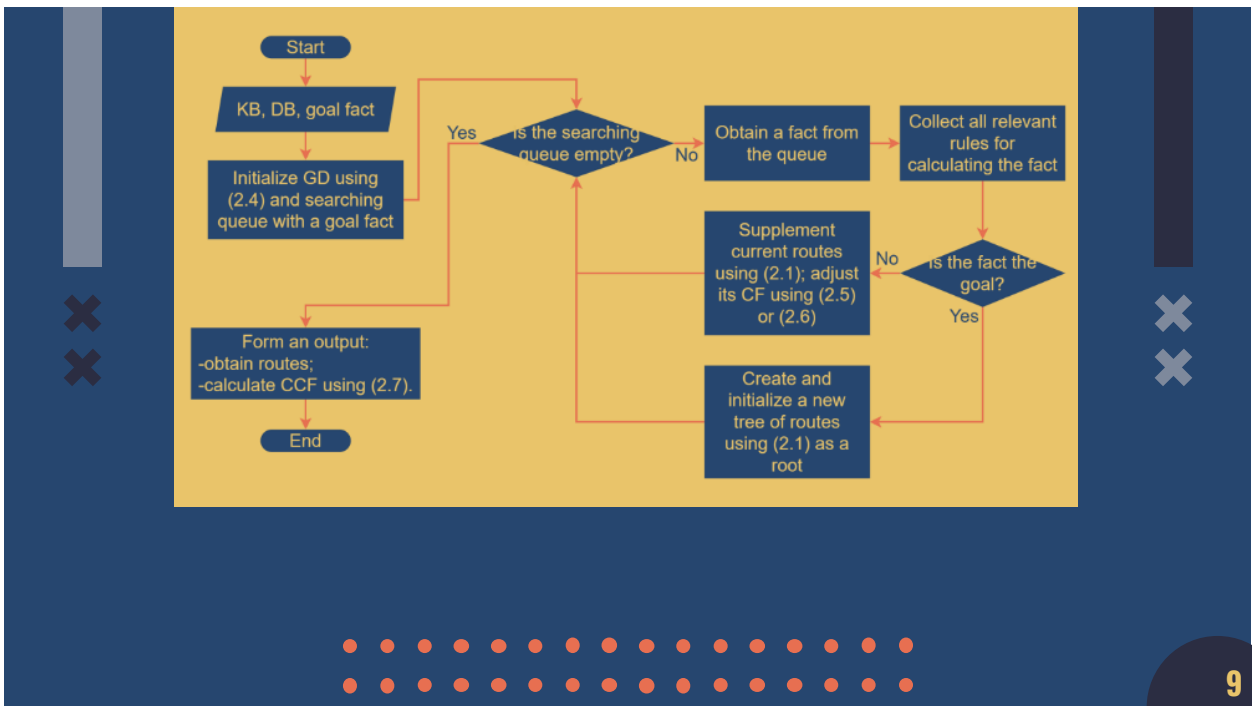


$$\{ \langle f_j, \{r_k, \forall r_k \in KB_f\} \rangle, \forall f_j \in F \} \quad (2.4)$$

де  $KB_f$  – множина правил для визначення факту  $f$ ;

$F$  – множиною всіх фактів, що використовуються у системі.





$$cf(H, E_1 \cap E_2 \cap \dots \cap E_n) = \min[cf(E_1), cf(E_2), \dots, cf(E_n)] * cf \quad (2.5)$$

$$cf(H, E_1 \cup E_2 \cup \dots \cup E_n) = \max[cf(E_1), cf(E_2), \dots, cf(E_n)] * cf \quad (2.6)$$

$$cf(cf_1, cf_2) = \begin{cases} cf_1 + cf_2 * (1 - cf_1) & cf_1 > 0 \text{ and } cf_2 > 0 \\ \frac{cf_1 + cf_2}{1 - \min[|cf_1|, |cf_2|]} & cf_1 < 0 \text{ or } cf_2 < 0 \\ cf_1 + cf_2 * (1 + cf_1) & cf_1 < 0 \text{ and } cf_2 < 0 \end{cases} \quad (2.7)$$

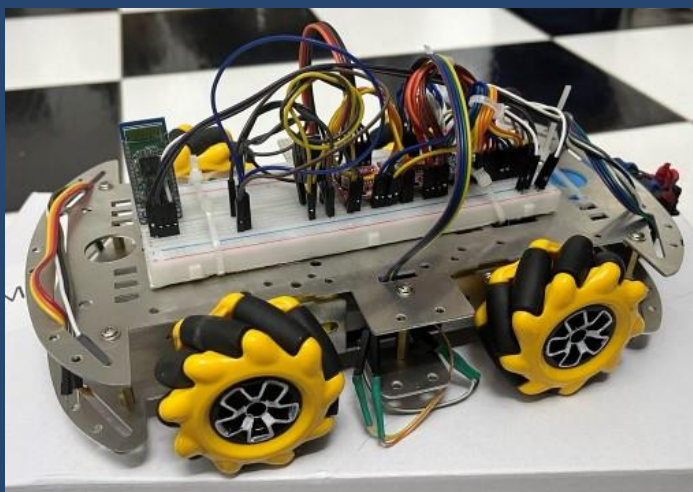
GetLeaves

RestorePath

UpdateAncestors

11

## Апаратна модель прототипу



12



**Мікроконтролер  
Arduino Mega 2560**



**WiFi модуль  
ESP8266-01**



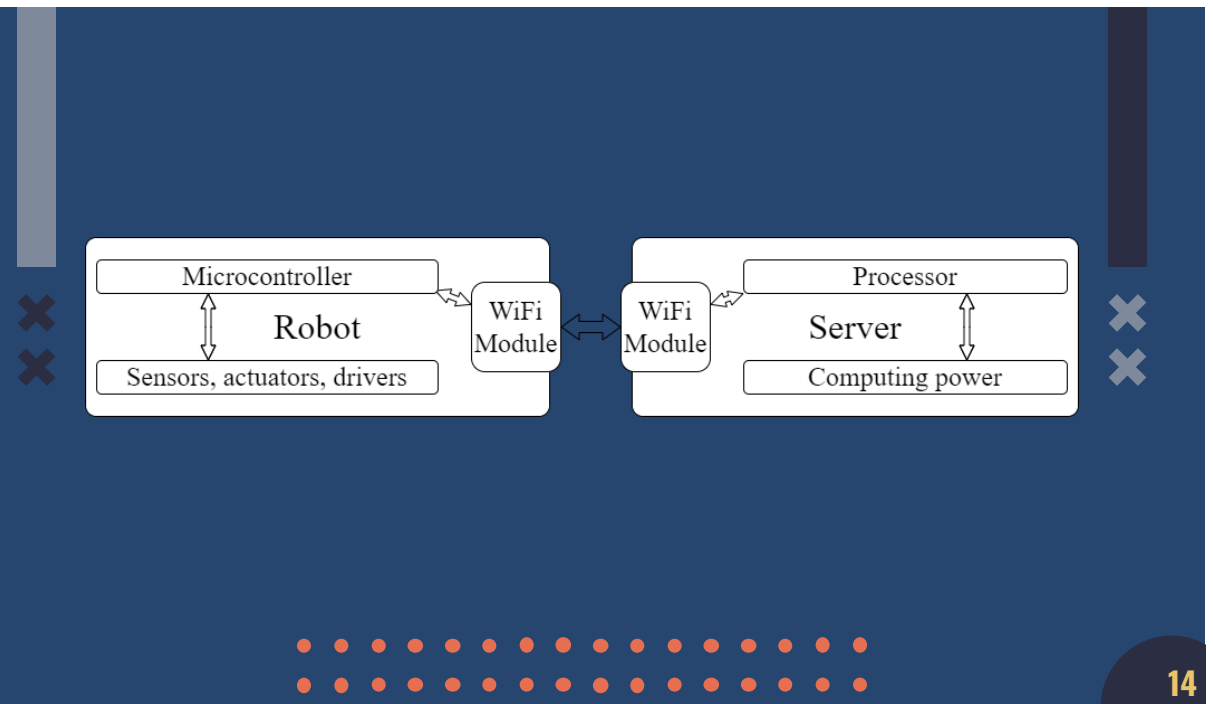
**Плата розширення  
Arduino Motor Shield**

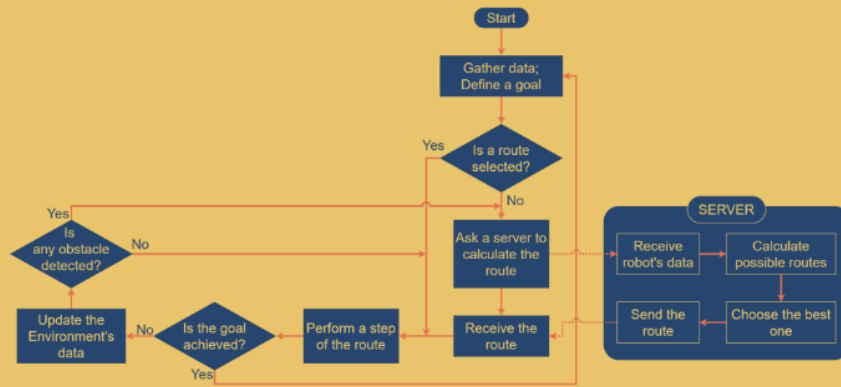


**Датчик відображення  
KY-033**



**Ультразвуковий  
датчик HC-SR04**





## Алгоритм непрерывного планирования

15

## АПРОБАЦІЯ РЕЗУЛЬТАТІВ

Control, Navigation and Communication Systems. 2024. No. 2

ISSN 2073-7394

УДК 004.89

doi: 10.26906/SUNZ.2024.2.106

А. О. Каргін<sup>1,2</sup>, Д. О. Гієвський<sup>2</sup>, Д. Г. Олійник<sup>1</sup><sup>1</sup>Харківський національний університет радіоелектроніки, Харків, Україна<sup>2</sup>Український державний університет залізничного транспорту, Харків, Україна

### БЕЗПЕРЕРВНЕ ПЛАНУВАННЯ І СИТУАЦІЙНЕ УПРАВЛІННЯ ЯК ЗАВДАННЯ ШТУЧНОГО ІНТЕЛЕКТУ ЩО ВІДЧУВАЄ

**Анотація.** Мотивація. Незважаючи на значний прогрес у галузі створення Безлюдних Систем (БС), забезпечення необхідного рівня їх автономності залишається актуальним завданням. Важливу роль у його вирішенні відіграє штучний інтелект. Особливості безлюдних систем породили нову модель «Штучний інтелект що відчуває» (ШІВ), що підтримує автономність. Метою даної роботи є розробка алгоритму що підтримує модель безперервного планування і ситуаційного управління імплементовану у систему Goal-Directed Control ШІВ. Предметом дослідження є методи та моделі управління автономними мобільними роботами на підставі даних від різноманітних сенсорів. Результати. Особливістю завдань управління автономними мобільними роботами є те, що для прийняття рішення в режимі реального часу щодо поточних дій використовують стан виконання плану, поточну ситуацію та можливість виконання частини плану дій, що залишилася, для досягнення мети. Наведено структуру багаторівневої розподіленої системи нечітких правил управління у поєднанні з системою продукційних правил планування. Розглянуто модифікований механізм нечіткого висновку, який завдяки введенню фактора впевненості здатний обробляти як факти щодо стану оточення так і стану виконання плану. Наведено алгоритм безперервного планування й приклади контрольних розрахунків. Висновок. Показано, що модифікація традиційного механізму логічного виведення в нечітких логічних системах, по-перше, шляхом введення контекстної пам'яті, що містить контекстні факти, по-друге, уведення стану фактів, як і значень входних змінних у вигляді нечітких факторів впевненості, дозволяє застосувати їх для управління БС та використовувати усі переваги нечіткого управління щодо обробки невизначеності. Розробка традиційної нечіткої системи, спрямованої на управління реалізацією плану дій для автономного робота з урахуванням висезазначених умов, що характеризується значною кількістю входних числових змінних від датчиків, є нерозв'язним завданням. Запропонована модель, яка складається з компонентів двох типів систем нечітких систем з лінгвістичними змінними та нечітких продукційних систем з факторами впевненості дозволяє перелічити проблеми та зберегає переваги традиційних нечітких систем у поєднанні з невизначеністю. Напрямок подальших досліджень. Майбутні кроки можуть бути присвячено розвитку моделі та інструментарію для інших класів безлюдних систем.

**Ключові слова:** автономні інтелектуальні безлюдні системи, штучний інтелект що відчуває, безперервне планування, ситуаційне управління.

DOI : <https://doi.org/10.26906/SUNZ.2024.2.106>

16

## ВИСНОВКИ

### АЛГОРИТМ КОМБІНОВАНОГО ЗВОТНОГО ВИВОДУ

- алгоритм приймає у якості вхідних даних: БЗ у форматі правил «ЯКЩО-ТО», БД із відомими фактами, кінцевий факт як ціль пошуку;
- алгоритм відстежує знайдені проміжні факти для зменшення кількості обчислень;
- алгоритм уточнює ФВ шляхів у процесі їх доповнення знайденими необхідними фактами;
- алгоритм має можливість відновлення шляху, якщо вже були знайдені відповідні правила;
- результатом роботи алгоритму є множина шляхів для розрахунку цільового факту з розрахованими ФВ кожного шляху та КФВ.

### ОПИС АПАРАТНОЇ ТА ПРОГРАМНОЇ МОДЕЛІ ПРОТОТИПУ СИСТЕМИ

- компоненти апаратної частини моделі прототипу системи;
- архітектуру апаратної частини моделі прототипу системи;
- архітектуру та компоненти програмної частини прототипу системи;
- алгоритм неперервного планування;
- шаблон програмного управління АІБС.

## ДОДАТОК Б

## База знань та приклад роботи алгоритму

## Б.1 Приклад роботи алгоритму

R <sub>1</sub> :	ЯКЩО	*A <sup>Load</sup> TA *Move <sup>F</sup> <sub>1</sub>	ТО	*1 <sup>Move</sup> (cf=0.9);
R <sub>2</sub> :	ЯКЩО	*1 <sup>Move</sup> TA *Move <sup>LF</sup> <sub>2</sub>	ТО	*4 <sup>Move</sup> (cf=0.9);
R <sub>3</sub> :	ЯКЩО	*4 <sup>Move</sup> TA *Move <sup>RF</sup> <sub>3</sub>	ТО	*5 <sup>Move</sup> (cf=0.9);
R <sub>4</sub> :	ЯКЩО	*5 <sup>Move</sup> TA *Move <sup>F</sup> <sub>4</sub>	ТО	*B <sup>Unload</sup> (cf=0.9);
R <sub>5</sub> :	ЯКЩО	*A <sup>Load</sup> TA u <sup>3</sup> <sub>1</sub>	ТО	*B <sup>Unload</sup> (cf=0.7);
R <sub>6</sub> :	ЯКЩО	*1 <sup>Move</sup> TA u <sup>2</sup> <sub>1</sub>	ТО	*B <sup>Unload</sup> (cf=0.75);
R <sub>7</sub> :	ЯКЩО	*4 <sup>Move</sup> TA u <sup>1</sup> <sub>1</sub>	ТО	*B <sup>Unload</sup> (cf=0.8);
R <sub>8</sub> :	ЯКЩО	*A <sup>Load</sup> TA *Move <sup>F</sup> <sub>1</sub>	ТО	*1 <sup>Move</sup> (cf=0.9);
R <sub>9</sub> :	ЯКЩО	*1 <sup>Move</sup> TA *Move <sup>F</sup> <sub>5</sub>	ТО	*2 <sup>Move</sup> (cf=0.9);
R <sub>10</sub> :	ЯКЩО	*2 <sup>Move</sup> TA *Move <sup>RF</sup> <sub>6</sub>	ТО	*C <sup>Charge</sup> (cf=0.9);
R <sub>11</sub> :	ЯКЩО	*A <sup>Load</sup> TA u <sup>2</sup> <sub>2</sub>	ТО	*C <sup>Charge</sup> (cf=0.8);
R <sub>12</sub> :	ЯКЩО	*1 <sup>Move</sup> TA u <sup>1</sup> <sub>2</sub>	ТО	*C <sup>Charge</sup> (cf=0.85);
R <sub>13</sub> :	ЯКЩО	*A <sup>Load</sup> TA *Move <sup>F</sup> <sub>1</sub>	ТО	*1 <sup>Move</sup> (cf=0.9);
R <sub>14</sub> :	ЯКЩО	*1 <sup>Move</sup> TA *Move <sup>F</sup> <sub>5</sub>	ТО	*2 <sup>Move</sup> (cf=0.9);
R <sub>15</sub> :	ЯКЩО	*2 <sup>Move</sup> TA *Move <sup>F</sup> <sub>7</sub>	ТО	*3 <sup>Move</sup> (cf=0.9);
R <sub>16</sub> :	ЯКЩО	*3 <sup>Move</sup> TA *Move <sup>RF</sup> <sub>8</sub>	ТО	*D <sup>Charge</sup> (cf=0.9);
R <sub>17</sub> :	ЯКЩО	*A <sup>Load</sup> TA u <sup>3</sup> <sub>3</sub>	ТО	*D <sup>Charge</sup> (cf=0.7);
R <sub>18</sub> :	ЯКЩО	*1 <sup>Move</sup> TA u <sup>2</sup> <sub>3</sub>	ТО	*D <sup>Charge</sup> (cf=0.75);
R <sub>19</sub> :	ЯКЩО	*2 <sup>Move</sup> TA u <sup>1</sup> <sub>3</sub>	ТО	*D <sup>Charge</sup> (cf=0.85);
R <sub>20</sub> :	ЯКЩО	*C <sup>Charge</sup> TA *Move <sup>F</sup> <sub>2</sub>	ТО	*2 <sup>Move</sup> (cf=0.9);
R <sub>21</sub> :	ЯКЩО	*2 <sup>Move</sup> TA *Move <sup>F</sup> <sub>7</sub>	ТО	*3 <sup>Move</sup> (cf=0.9);
R <sub>22</sub> :	ЯКЩО	*3 <sup>Move</sup> TA *Move <sup>LF</sup> <sub>9</sub>	ТО	*5 <sup>Move</sup> (cf=0.9);
R <sub>23</sub> :	ЯКЩО	*5 <sup>Move</sup> TA *Move <sup>RF</sup> <sub>10</sub>	ТО	*B <sup>Unload</sup> (cf=0.9);
R <sub>24</sub> :	ЯКЩО	*C <sup>Charge</sup> TA u <sup>3</sup> <sub>4</sub>	ТО	*B <sup>Unload</sup> (cf=0.75);
R <sub>25</sub> :	ЯКЩО	*2 <sup>Move</sup> TA u <sup>2</sup> <sub>4</sub>	ТО	*B <sup>Unload</sup> (cf=0.8);
R <sub>26</sub> :	ЯКЩО	*3 <sup>Move</sup> TA u <sup>1</sup> <sub>4</sub>	ТО	*B <sup>Unload</sup> (cf=0.85);
R <sub>27</sub> :	ЯКЩО	*D <sup>Charge</sup> TA *Move <sup>F</sup> <sub>11</sub>	ТО	*3 <sup>Move</sup> (cf=0.9);
R <sub>28</sub> :	ЯКЩО	*3 <sup>Move</sup> TA *Move <sup>F</sup> <sub>12</sub>	ТО	*5 <sup>Move</sup> (cf=0.9);
R <sub>29</sub> :	ЯКЩО	*5 <sup>Move</sup> TA *Move <sup>RF</sup> <sub>9</sub>	ТО	*B <sup>Unload</sup> (cf=0.9);
R <sub>30</sub> :	ЯКЩО	*D <sup>Charge</sup> TA u <sup>2</sup> <sub>5</sub>	ТО	*B <sup>Unload</sup> (cf=0.75);
R <sub>31</sub> :	ЯКЩО	*3 <sup>Move</sup> TA u <sup>1</sup> <sub>5</sub>	ТО	*B <sup>Unload</sup> (cf=0.8). (Б.1)

## Лістинг Б.1 – Тестування роботи алгоритму

**Input** KB that consists of (5) and (6); DB that consists of  $^*2^{Move}$  with its  $cf = 0.9$ ; goal is  $^*5^{Move}$

1)  $queue = \{^*5^{Move}\}$ ;  $tree\_container = \{\}$ ;  $GD = \{\langle ^*2^{Move}, \{0\}\rangle, \langle ^*5^{Move}, \{\}\rangle\}$ ;

2)  $fact = ^*5^{Move}$ ;  $rule\_container = \{3, 22, 28\}$   
 $queue = \{^*4^{Move}, ^*3^{Move}\}$ ;  
 $tree\_container = [$   
 $tree\_1 = \{\langle N = 0, R_N = 3, P_N = null, C_N = \{\}, D_N = \{\langle ^*4^{Move}, null\rangle\}, CF_N = 0.9, CF_B = 0.9\rangle\}$ ;  
 $tree\_2 = \{\langle 1, 22, null, \{\}, \{\langle ^*3^{Move}, null\rangle\}, 0.9, 0.9\rangle\}$ ;  
 $tree\_3 = \{\langle 2, 28, null, \{\}, \{\langle ^*3^{Move}, null\rangle\}, 0.9, 0.9\rangle\}$ ;  
 $GD = \{\langle ^*2^{Move}, \{0\}\rangle, \langle ^*5^{Move}, \{3, 22, 28\}\rangle, \langle ^*4^{Move}, \{\}\rangle, \langle ^*3^{Move}, \{\}\rangle\}$ ;

3)  $fact = ^*4^{Move}$ ;  $rule\_container = \{2\}$   
 $queue = \{^*3^{Move}, ^*1^{Move}\}$ ;  
 $tree\_container = [$   
 $\{\langle 0, 3, null, \{3\}, \{\langle ^*4^{Move}, \{\langle 2, 0.9\rangle\}\rangle\}, 0.9, null\rangle,$   
 $\langle 3, 2, 0, \{\}, \{\langle ^*1^{Move}, null\rangle\}, 0.9, 0.9\rangle\}$ ;  
 $\{\langle 1, 22, null, \{\}, \{\langle ^*3^{Move}, null\rangle\}, 0.9, 0.9\rangle\}$ ;  
 $\{\langle 2, 28, null, \{\}, \{\langle ^*3^{Move}, null\rangle\}, 0.9, 0.9\rangle\}$ ;  
 $GD = \{\langle ^*2^{Move}, \{0\}\rangle, \langle ^*5^{Move}, \{3, 22, 28\}\rangle, \langle ^*4^{Move}, \{2\}\rangle, \langle ^*3^{Move}, \{\}\rangle\}$ ;

4)  $fact = ^*3^{Move}$ ;  $rule\_container = \{15, 21, 27\}$   
 $queue = \{^*1^{Move}, ^*D^{Charge}\}$ ;  
 $tree\_container = [$   
 $\{\langle 0, 3, null, \{3\}, \{\langle ^*4^{Move}, \{\langle 2, 0.9\rangle\}\rangle\}, 0.9, null\rangle,$   
 $\langle 3, 2, 0, \{\}, \{\langle ^*1^{Move}, null\rangle\}, 0.9, 0.9\rangle\}$ ;  
 $\{\langle 1, 22, null, \{4, 5, 6\}, \{\langle ^*3^{Move}, \{\langle 15, 0.81\rangle, \langle 21, 0.81\rangle, \langle 27, 0.9\rangle\}\rangle\},$   
 $0.9, null\rangle,$   
 $\langle 4, 15, 1, \{\}, \{\langle ^*2^{Move}, \{\langle 0, 0.9\rangle\}\rangle\}, 0.81, 0.729\rangle,$   
 $\langle 5, 21, 1, \{\}, \{\langle ^*2^{Move}, \{\langle 0, 0.9\rangle\}\rangle\}, 0.81, 0.729\rangle,$   
 $\langle 6, 27, 1, \{\}, \{\langle ^*D^{Charge}, null\rangle\}, 0.9, 0.9\rangle\}$ ;  
 $\{\langle 2, 28, null, \{7, 8, 9\}, \{\langle ^*3^{Move}, \{\langle 15, 0.81\rangle, \langle 21, 0.81\rangle, \langle 27, 0.9\rangle\}\rangle\},$   
 $0.9, null\rangle.$   
 $\langle 7, 15, 1, \{\}, \{\langle ^*2^{Move}, \{\langle 0, 0.9\rangle\}\rangle\}, 0.81, 0.729\rangle,$   
 $\langle 8, 21, 1, \{\}, \{\langle ^*2^{Move}, \{\langle 0, 0.9\rangle\}\rangle\}, 0.81, 0.729\rangle,$   
 $\langle 9, 27, 1, \{\}, \{\langle ^*D^{Charge}, null\rangle\}, 0.9, 0.9\rangle\}$ ;

$GD = \{\langle ^*2^{Move}, \{0\}\rangle, \langle ^*5^{Move}, \{3, 22, 28\}\rangle, \langle ^*4^{Move}, \{2\}\rangle,$   
 $\langle ^*3^{Move}, \{15, 21, 27\}\rangle, \langle ^*1^{Move}, \{\}\rangle, \langle ^*D^{Charge}, \{\}\rangle\}$ ;

5)  $fact = ^*1^{Move}$ ;  $rule\_container = \{1, 8, 13\}$   
 $queue = \{^*D^{Charge}, ^*A^{Load}\}$ ;

```

tree_container = [
{<0,3, null,{3}, {<*4Move,{<2,0.9>>}},0.9, null>,
  <3, 2, 0, {10,11,12}, {<*7Move, ,{<1,0.9>,<8,0.9>,<13,0.9>>}},
  0.9, null >,
  <10, 1, 3, {}, {<*ALoad,null>}, 0.9, 0.9>,
  <11, 8, 3, {}, {<*ALoad,null>}, 0.9, 0.9>,
  <12, 13, 3, {}, {<*ALoad,null>}, 0.9, 0.9>},
{<1, 22, null, {4,5,6}, {<*3Move,{<15,0.81>,<21,0.81>,<27,0.9>>}},
  0.9, null>,
  <4, 15, 1, {}, {<*2Move,{<0,0.9>>}}, 0.81, 0.729>,
  <5, 21, 1, {}, {<*2Move,{<0,0.9>>}}, 0.81, 0.729>,
  <6, 27, 1, {}, {<*DCharge,null>}, 0.9, 0.9>},
{< 2, 28, null, {7,8,9}, {<*3Move,{<15,0.81>,<21,0.81>,<27,0.9>>}},
  0.9, null>.
  <7, 15, 1, {}, {<*2Move,{<0,0.9>>}}, 0.81, 0.729>,
  <8, 21, 1, {}, {<*2Move,{<0,0.9>>}}, 0.81, 0.729>,
  <9, 27, 1, {}, {<*DCharge,null>}, 0.9, 0.9>},,];
GD = {<*2Move,{0}>, <*5Move,{3,22,28}>, <*4Move,{2}>,
  <*3Move,{15,21,27}>, <*7Move,{1,8,13}>, <*DCharge,{}>};
6) fact = *DCharge; rule_container = {17,18,19}
queue = {*ALoad};
tree_container = [
{<0,3, null,{3}, {<*4Move,{<2,0.9>>}},0.9, null>,
  <3, 2, 0, {10,11,12}, {<*7Move,{<1,0.9>,<8,0.9>,<13,0.9>>}}, 0.9,
  null >,
  <10, 1, 3, {}, {<*ALoad,null>}, 0.9, 0.9>,
  <11, 8, 3, {}, {<*ALoad,null>}, 0.9, 0.9>,
  <12, 13, 3, {}, {<*ALoad,null>}, 0.9, 0.9>},
{<1,
  22,
  null,
  {4,5,6},
  {<*3Move,{<15,0.81>,<21,0.81>,<27,0.6885>>}}, 0.61965, null>,
  <4, 15, 1, {}, {<*2Move,{<0,0.9>>}}, 0.81, 0.729>,
  <5, 21, 1, {}, {<*2Move,{<0,0.9>>}}, 0.81, 0.729>,
  <6,
  27,
  1,
  {13,14,18},
  {<*DCharge,
  {<17,0.7>,<18,0.75>,<19,0.61965>>}}, 0.6885, null>},
  <13, 17, 6, {}, {<*ALoad,null>}, 0.7, 0.9>
  <14, 18, 6, {15,16,17}, {<*7Move,{<1,0.9>,<8,0.9>,<13,0.9>>}},
  0.9, null>
  <15, 1, 14, {}, {<*ALoad,null>}, 0.9, 0.9>,
  <16, 8, 14, {}, {<*ALoad,null>}, 0.9, 0.9>,
  <17, 13, 14, {}, {<*ALoad,null>}, 0.9, 0.9>,
  <18, 19, 6, {}, {<*2Move,{<0,0.9>>}}, 0.795, 0.61965>},
{<
  2,
  28,
  null,
  {7,8,9},
  {<*3Move,{<15,0.81>,<21,0.81>,<27,0.6885>>}}, 0.61965, null >.
  <7, 15, 1, {}, {<*2Move,{<0,0.9>>}}, 0.81, 0.729>,

```

```

<8, 21, 1, {}, {<*2Move, {<0,0.9>>}}, 0.81, 0.729>,
<9,          27,          1,          {19,20,24},          {<*DCharge,
{<17,0.7>,<18,0.75>,<19,0.61965>>}}, 0.6885, null>},
<19, 17, 6, {}, {<*ALoad, null>}}, 0.7, 0.9>
<20, 18, 6, {21,22,23}, {<*7Move, {<1,0.9>,<8,0.9>,<13,0.9>>}},
0.9, null>
<21, 1, 14, {}, {<*ALoad, null>}}, 0.9, 0.9>,
<22, 8, 14, {}, {<*ALoad, null>}}, 0.9, 0.9>,
<23, 13, 14, {}, {<*ALoad, null>}}, 0.9, 0.9>,
<24, 19, 6, {}, {<*2Move, {<0,0.9>>}}, 0.795, 0.61965>}];
GD = {<*2Move, {0}>, <*5Move, {3,22,28}>, <*4Move, {2}>,
<*3Move, {15,21,27}>, <*7Move, {1,8,13}>, <*DCharge, {17,18,19}>};
7) Found routes: {{1>2>3}, {8>2>3}, {13>2>3}, {15>22}, {21>22},
{17>27>22}, {1>18>27>22}, {8>18>27>22}, {13>18>27>22},
{19>27>22}, {15>28}, {21>28}, {17>27>28}, {1>18>27>28},
{8>18>27>28}, {13>18>27>28}, {19>27>28}}
Output Valid_routes:      {{15>22}(0.729),      {21>22}(0.729),
{19>27>22}(0.61965),      {15>28}(0.729),      {21>28}(0.729),
{19>27>28}(0.61965)}
Combined_certainty_factor = 0.9992

```