

Sztuczna Inteligencja i Inżynieria Wiedzy

Lab 3

Autor: Jakub Szwedowicz 243416

Prowadzący: dr inż. Piotr Syga

Zajęcia: czw 7:30

Spis treści

1. Konstrukcja prostej sieci semantycznej opisującej wybrane urządzenie lub aplikację: 20 pkt.....	3
2. Pokazanie możliwości wnioskowania i działanie unifikacji na kilku dobrze dobranych przykładach zapytań i wnioskowaniu: 10 pkt.	3
3. Wydobycie z treści instrukcji zdań opisujących reguły diagnostyczne, zasady postępowania, sposoby postępowanie w razie występowania problemów itd.: 10pkt.....	4
4. Opracowanie bazy wiedzy w Prologu (bazy klauzul) opisującej wiedzę wyrażoną w tych zdaniach: 30 pkt.....	6
5. Pokazanie możliwości opracowanej bazy wiedzy na kilku przykładowych problemach rozwiązanych za pomocą wnioskowania (minimum 3): 30 pkt.....	9
6. Wnioski	10

1. Konstrukcja prostej sieci semantycznej opisującej wybrane urządzenie lub aplikację: 20 pkt.

W liście zadań wykorzystana zostanie struktura serwera od firmy [dell R720](#).

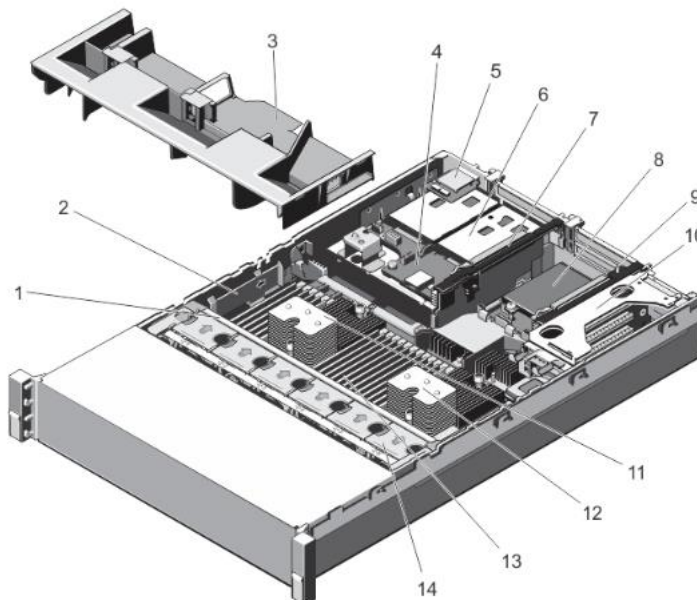


Figure 15. Inside the System—PowerEdge R720xd

- | | | | |
|----|---------------------------|----|-----------------------------|
| 1 | cooling-fan assembly | 2 | cable securing bracket |
| 3 | cooling shroud | 4 | hard-drive backplane (back) |
| 5 | vFlash media slot | 6 | hard drives (back) (2) |
| 7 | expansion-card riser 3 | 8 | network daughter card |
| 9 | expansion-card riser 2 | 10 | expansion-card riser 1 |
| 11 | heat sink for processor 1 | 12 | heat sink for processor 2 |
| 13 | DIMMs (24) | 14 | cooling fans (6) |

2. Pokazanie możliwości wnioskowania i działanie unifikacji na kilku dobrze dobranych przykładach zapytań i wnioskowaniu: 10 pkt.

Wnioskowanie w Prologu odnosi się do procesu dedukcji, w którym system Prolog próbuje znaleźć odpowiedzi na zadane pytania, korzystając z dostępnej wiedzy i zdefiniowanych zasad.

Wnioskowanie w Prologu opiera się na mechanizmie przeszukiwania przestrzeni możliwości, aby odnaleźć odpowiednie rozwiązania. Prolog wykorzystuje algorytm unifikacji do dopasowania zapytań do faktów i reguł w bazie wiedzy.

Unifikacja w Prologu to proces dopasowania wyrażeń, które mogą składać się z zmiennych, stałych i struktur danych. Głównym celem unifikacji jest znalezienie wartości zmiennych, które spowodują, że dwa wyrażenia staną się identyczne lub będą ze sobą "unifikowalne". Jeśli dwa wyrażenia są unifikowalne, to oznacza, że mogą zostać dopasowane do siebie, co prowadzi do sukcesu wnioskowania.

3. Wydobycie z treści instrukcji zdań opisujących reguły diagnostyczne, zasady postępowania, sposoby postępowanie w razie występowania problemów itd.: 10pkt.

Instrukcja serwera zawiera spis problemów, które mogą wystąpić oraz potencjalne sposoby rozwiązania problemów. Na podstawie poniższego zdjęcia wybrano następujące problemy:

Problemy:

- System Memory (w prologu: memory_failure),
- An Internal USB Key (prolog: internal_usb_key_failure),
- An SD Card (prolog: sd_card_failure),
- An Optical Drive (prolog: optical_drive_failure),
- A Tape Backup Unit (prolog: tape_backup_unit_failure),
- A Hard Drive (prolog: hard_drive_failure)

Troubleshooting Your System

Safety First—For You And Your System

CAUTION: Many repairs may only be done by a certified service technician. You should only perform troubleshooting and simple repairs as authorized in your product documentation, or as directed by the online or telephone service and support team. Damage due to servicing that is not authorized by Dell is not covered by your warranty. Read and follow the safety instructions that came with the product.

Topics:

- Troubleshooting System Startup Failure
- Troubleshooting External Connections
- Troubleshooting The Video Subsystem
- Troubleshooting A USB Device
- Troubleshooting A Serial I/O Device
- Troubleshooting A NIC
- Troubleshooting A Wet System
- Troubleshooting A Damaged System
- Troubleshooting The System Battery
- Troubleshooting Power Supplies
- Troubleshooting Cooling Problems
- Troubleshooting Cooling Fans
- Troubleshooting System Memory
- Troubleshooting An Internal USB Key
- Troubleshooting An SD Card
- Troubleshooting An Optical Drive
- Troubleshooting A Tape Backup Unit
- Troubleshooting A Hard Drive
- Troubleshooting A Storage Controller
- Troubleshooting Expansion Cards
- Troubleshooting Processors

Na przykładzie awarii pamięci można przeczytać, że do kroków diagnostycznych należą między innymi:

Awaria pamięci (memory_failure):

- (1) Jeśli system jest w stanie funkcjonować z błędami (system_unstable) to użytkownik powinien przeprowadzić test diagnostyczny (run_diagnostic_test).
- (2) Jeśli system nie jest w stanie funkcjonować (system_does_not_boot), (4) sprawdź ustawienia przy starcie systemu i (7) zweryfikuj czy problemem konfiguracja pamięci (check_memory_channels). (8) Jeśli to nic nie da to użytkownik może również spróbować resetu ustawień pamięci (reset_memory_modules)
- (3, 13) Jeśli na ekranie wyświetla się komunikat o błędzie (error_message_in_screen) lub (12) tekst diagnostyczny wskazuje na problem z pamięcią (memory_error_detected) to być może problemem jest gniazdo DIMM (dimm_socket) i rozwiązaniem będzie wymiana modułów pamięci (replace_memory_module)

Troubleshooting System Memory

CAUTION: Many repairs may only be done by a certified service technician. You should only perform troubleshooting and simple repairs as authorized in your product documentation, or as directed by the online or telephone service and support team. Damage due to servicing that is not authorized by Dell is not covered by your warranty. Read and follow the safety instructions that came with the product.

- 1 If the system is operational, run the appropriate diagnostic test. See Using System Diagnostics for available diagnostic tests. If diagnostics indicates a fault, follow the corrective actions provided by the diagnostic program.
- 2 If the system is not operational, turn off the system and attached peripherals, and unplug the system from the power source. Wait at least 10 seconds and then reconnect the system to power.
- 3 Turn on the system and attached peripherals and note the messages on the screen. If an error message is displayed indicating a fault with a specific memory module, go to step 12.
- 4 Enter the System Setup and check the system memory setting. Make any changes to the memory settings, if needed. If the memory settings match the installed memory but a problem is still indicated, go to step 12.
- 5 Turn off the system and attached peripherals, and disconnect the system from the electrical outlet.
- 6 Open the system.
- 7 Check the memory channels and ensure that they are populated correctly.
- 8 Reseat the memory modules in their sockets.

Troubleshooting Your System 121

- 9 Close the system.
- 10 Enter the System Setup and check the system memory setting. If the problem is not resolved, proceed with the next step.
- 11 Open the system.
- 12 If a diagnostic test or error message indicates a specific memory module as faulty, swap or replace the module with a known good memory module.
- 13 To troubleshoot an unspecified faulty memory module, replace the memory module in the first DIMM socket with a module of the same type and capacity. If an error message is displayed on the screen, this may indicate a problem with the installed DIMM type(s), incorrect DIMM installation, or defective DIMM(s). Follow the on-screen instructions to resolve the problem. For more information, see General Memory Module Installation Guidelines.
- 14 Close the system.
- 15 As the system boots, observe any error message that is displayed and the diagnostic indicators on the front of the system.
- 16 If the memory problem is still indicated, repeat step 12 through step 15 for each memory module installed.

Na podstawie opisanych kroków można wywnioskować następującą strukturę:

```

25 % Potencjalne problemy i symptomy jakie moga wystąpić w serwerze
26 problem(memory_failure, memory_error_detected).
27 problem(memory_failure, system_does_not_boot).
28 problem(memory_failure, system_unstable).
29 problem(memory_failure, error_message_in_screen).

45 % Potencjalne czynności jakie trzeba wykonać w celu rozwiązania problemu
46 solution(memory_failure, run_diagnostic_test).
47 solution(memory_failure, check_memory_channels).
48 solution(memory_failure, reset_memory_modules).
49 solution(memory_failure, replace_memory_module).

```

4. Opracowanie bazy wiedzy w Prologu (bazy klauzul) opisującej wiedzę wyrażoną w tych zdaniach: 30 pkt

Struktura serwera może przyjmować postać drzewa, gdzie korzeniem jest sam serwer. W następnej kolejności można mówić o jego peryferiach (np. procesor), który składa się z pewnych elementów, np. rdzenia. Poszczególne peryferia mają wpływ na niektóre cechy systemu, np. rodzaj procesora wpływa na prędkość taktowania.

Serwer i peryferia:

```

1 % Początek drzewa struktury serwera. Server - podzespół
2 component(server, processor).
3 component(server, memory).
4 component(server, hard_drive).
5 component(server, optical_drive).
6 component(server, tape_backup_unit).
7 component(server, internal_usb_key).
8 component(server, sd_card).
9 component(server, storage_controller).
10 component(server, expansion_card).

```

Peryferia i podzespoły:

```

12 % Elementy podzespołów
13 component(processor, core).
14 component(memory, dimm_socket).
15 component(hard_drive, platter).
16 component(hard_drive, read_write_head).
17 component(optical_drive, laser).
18 component(optical_drive, motor).
19 component(tape_backup_unit, tape_cartridge).
20 component(internal_usb_key, usb_connector).
21 component(sd_card, sd_card_slot).
22 component(storage_controller, controller_card).
23 component(expansion_card, expansion_card_slot).

```

Peryferia i cechy na które mają wpływ:

```

59 % Cechy poszczególnych peryferiów.
60 % Wskazują na parametry serwera na które mają wpływ poszczególne peryferia.
61 % Np. na dostępne interfejsy urządzenia wpływają storage_controller oraz expansion_card
62 property(processor, clock_speed).
63 property(memory, capacity).
64 property(hard_drive, storage_capacity).
65 property(optional_drive, read_speed).
66 property(tape_backup_unit, storage_capacity).
67 property(internal_usb_key, storage_capacity).
68 property(sd_card, storage_capacity).
69 property(storage_controller, interace_type).
70 property(expansion_card, interface_type).

```

Dodatkowo zamodelowane są również problemy, które mogą wystąpić w systemie a także ich symptomy i rozwiązania. Problemy oczywiście mają również wpływ na niektóre podzespoły i ich elementy.

Problemy i symptomy:

```

25 % Potencjalne problemy i symptomy jakie mogą wystąpić w serwerze
26 problem(memory_failure, memory_error_detected).
27 problem(memory_failure, system_does_not_boot).
28 problem(memory_failure, system_unstable).
29 problem(memory_failure, error_message_in_screen).
30
31 problem(internal_usb_key_failure, usb_key_not_detected).
32 problem(internal_usb_key_failure, error_message_on_screen).
33
34 problem(sd_card_failure, sd_card_not_detected).
35 problem(sd_card_failure, error_message_on_screen).
36
37 problem(optical_drive_failure, optical_drive_not_detected).
38 problem(optical_drive_failure, error_message_on_screen).
39
40 problem(tape_backup_unit_failure, tape_backup_unit_not_detected).
41 problem(tape_backup_unit_failure, error_message_on_screen).
42
43 problem(hard_drive_failure, hard_drive_not_detected).

```

Problemy i rozwiązania:

```

45 % Potencjalne czynności jakie trzeba wykonać w celu rozwiązania problemu
46 solution(memory_failure, run_diagnostic_test).
47 solution(memory_failure, check_memory_channels).
48 solution(memory_failure, reset_memory_modules).
49 soultion(memory_failure, replace_memory_module).
50
51 solution(internal_usb_key_failure, enter_system_setup).
52 solution(internal_usb_key_failure, resear_usb_key).
53 solution(internal_usb_key_failure, check_usb_key_function).
54
55 solution(sd_card_failure, ensure_sd_card_port_enabled).
56 solution(sd_card_failure, replace_failed_sd_card).
57 solution(sd_card_failure, check_sd_card_function).

```

Problemy i podzespoły z ich elementami:

```

70 % Relacje opisujące który problem dotyczy którego peryferium lub jego elementu
71 affects(memory_failure, memory).
72 affects(internal_usb_key_failure, usb_connector).
73 affects(sd_card_failure, sd_card).
74 affects(optional_drive_failure, optical_drive).
75 affects(tape_backup_failure, tape_cartridge).

```


5. Pokazanie możliwości opracowanej bazy wiedzy na kilku przykładowych problemach rozwiązanych za pomocą wnioskowania (minimum 3): 30 pkt.

1. Odnajdywanie peryferii i komponentów powiązanych z problemem:


```

88 affects_components(Problem, Component) :-
89     affects(Problem, Component).
90 affects_components(Problem, Component) :-
91     affects(Problem, Perypherial),
92     component(Perypherial, Component).

```

Powyższe zapytania pozwalają wyznaczyć peryferia oraz elementy peryferii, które mogą być związane ze znanym problemem.

```

 affects_components(memory_failure, Component).

```

Component = memory

Component = dimm_socket

2. Wyznaczenie zbioru wszystkich problemów, które mogą przejawiać się danymi symptomami


```

95 possible_problems(Symptoms) :-
96     setof(Problem, Symptom^(member(Symptom, Symptoms), problem(Problem, Symptom)), Problems),
97     maplist(format("~w~n"), Problems).

```

Powyższe zapytanie najpierw iteruje po elementach listy, następnie matchuje jest z problemem. Zewnętrzna funkcja setof(+T, +G, -S) zwraca listę S złożoną z elementów o typie T, które spełniają G. Następnie elementy listy są drukowane w terminalu i oddzielone znakiem „\n”.

```

 possible_problems([error_message_on_screen,
memory_error_detected]).

```

```

internal_usb_key_failure
memory_failure
optical_drive_failure
sd_card_failure
tape_backup_unit_failure
true

```

3. Zwrócenie listy symptomów, które mogą dotyczyć danego peryferia:

```
100 symptoms_for_component(Component) :-
101     setof(Symptom, Problem^(affects(Problem, Component), problem(Problem, Symptom)), Symptoms),
102     maplist(format("~W~n"), Symptoms).
```

Powyższe zapytanie podobnie jak to z punktu wcześniej, matchuje Problem z Componentem i następnie wyznacza Symptom na podstawie Problem.



```
symptoms_for_component(sd_card).
```

```
error_message_on_screen
```

```
sd_card_not_detected
```

```
true
```

6. Wnioski

Celem ćwiczenia było poznanie sposobu reprezentowania wiedzy w symbolicznej formie przy użyciu prostych reguł i faktów. Wykorzystano język programowania Prolog i odpowiednie środowisko do tworzenia i testowania tych reguł.

Ćwiczenie pozwoliło zobaczyć, jak można zadawać pytania systemowi i uzyskiwać odpowiedzi na ich podstawie. Praktyczne zastosowania takiej reprezentacji wiedzy obejmują dziedziny jak sztuczna inteligencja i systemy ekspertowe.

Dzięki ćwiczeniu można było zrozumieć, jak działa wnioskowanie na podstawie takiej symbolicznej wiedzy. Wykorzystanie gotowych rozwiązań w języku Prolog ułatwiło eksplorację tego tematu, a ukończenie ćwiczenia dało podstawową wiedzę i umiejętności, które mogą być przydatne w dalszych badaniach lub praktyce.