**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](#_Toc136468525)

[2. Pokazanie możliwości wnioskowania i działanie unifikacji na kilku dobrze dobranych przykładach zapytań i wnioskowaniu: 10 pkt. 3](#_Toc136468526)

[3. Wydobycie z treści instrukcji zdań opisujących reguły diagnostyczne, zasady postępowania, sposoby postępowanie w razie występowania problemów itd.: 10pkt. 4](#_Toc136468527)

[4. Opracowanie bazy wiedzy w Prologu (bazy klauzul) opisującej wiedzę wyrażoną w tych zdaniach: 30 pkt 6](#_Toc136468528)

[5. Pokazanie możliwości opracowanej bazy wiedzy na kilku przykładowych problemach rozwiązanych za pomocą wnioskowania (minimum 3).: 30 pkt. 9](#_Toc136468529)

[6. Wnioski 10](#_Toc136468530)

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

Obraz zawierający tekst, elektronika, Inżynieria elektroniczna, zrzut ekranu

Opis wygenerowany automatycznieW liście zadań wykorzystana zostanie struktura serwera od firmy [dell R720](https://dl.dell.com/topicspdf/poweredge-r720_owners-manual_en-us.pdf).

# 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.

# 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)Obraz zawierający tekst, zrzut ekranu, Czcionka

  Opis wygenerowany automatycznie

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)
* Obraz zawierający tekst, zrzut ekranu, Czcionka, dokument

  Opis wygenerowany automatycznie(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)

Na podstawie opisanych kroków można wywnioskować następującą strukturę:

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

Obraz zawierający tekst, Czcionka, zrzut ekranu, linia

Opis wygenerowany automatycznie

# 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:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Peryferia i podzespoły:

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

Peryferia i cechy na które mają wpływ:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

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:

Obraz zawierający tekst, zrzut ekranu, Czcionka, dokument

Opis wygenerowany automatycznie

Problemy i rozwiązania:

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

Problemy i podzespoły z ich elementami:

Obraz zawierający tekst, Czcionka, zrzut ekranu

Opis wygenerowany automatycznie

# 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:

Obraz zawierający tekst, Czcionka, zrzut ekranu, biały

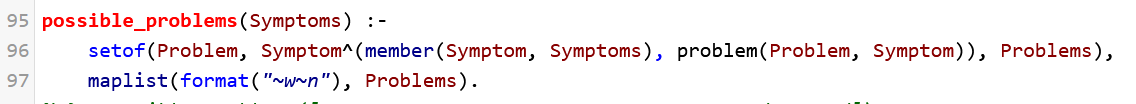
Opis wygenerowany automatycznie

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

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

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

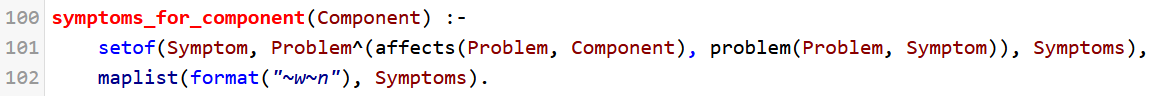


Powyższe zapytanie najpierw iteruje po elementach listy, następnie matchuje jest z problemem. Zewnętrzna funkcja setof(+T, +G, -S) zwraca listę S złożoną z elementów o typie T, które spełniają G. Następnie elementy listy są drukowane w terminalu i oddzielone znakiem „\n”.

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

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



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

Obraz zawierający tekst, zrzut ekranu, Czcionka, linia

Opis wygenerowany automatycznie

# 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.