

Sztuczna Inteligencja i Inżynieria Wiedzy

Sprawozdanie z zadania 3

Cel ćwiczenia

Celem ćwiczenia jest praktyczne zapoznanie się prezentacją wiedzy w postaci w oparciu o podzbiór logiki predykatów pierwszego rzędu oraz uruchamiania procesu wnioskowania w oparciu o zasadę rezolucji.

W ramach realizacji wybrano konkretne urządzenie oraz zapoznano się z jego instrukcją obsługi a szczególnie z rozdziałem o rozwiązywaniu problemów.

Wykonano konstrukcję prostej sieci semantycznej opisującej wybrane urządzenie.

Posiadając już w bazie wiedzy proste informacje na temat urządzenia pokazano możliwości wnioskowania informacji, które dotyczyły danego urządzenia.

Z rozdziału poświęconego rozwiązywaniu problemów wydobyto reguły diagnostyczne zasady postępowania oraz możliwości rozwiązania potencjalnych problemów. Zdobytą wiedzę przekształcono w dodatkowe klauzule i predykaty, których wywołanie pokazują możliwości opracowanej bazy wiedzy.

Technologia

Aby poprawnie wykonać zadanie użyto języka Prolog, oraz środowiska SWISH Prolog.

Implementacja i założenia

Urządzeniem, które zostało wybrane do realizacji tego zadania jest drukarka.

Drukarka to Brother DCP-T425W, stoi na komodzie w moim pokoju, a instrukcja obsługi była podstawą do zrealizowania tej listy. Instrukcja jest zatytułowana "Podręcznik szybkiej obsługi".

https://download.brother.com/welcome/doc101116/cv_dcp220_pol_fre_rus_qsg_a.pdf

Drukarka posiada takie elementy jak:

- kabel zasilający
- kabel USB
- taca na papier
- papier
- głowica drukująca
- tusz black
- tusz cyan
- tusz magenta
- tusz yellow

Drukarka posiada takie przyciski jak:

- przycisk włączania
- przycisk kopiowania w kolorze
- przycisk kopiowania w czerni



- przycisk szybkiego kopiowania
- przycisk drukowania w czerni

Drukarka posiada takie diody jak:

- dioda zasilania
- dioda ostrzegawcza
- dioda tuszu
- dioda wifi

Po zdobyciu tej wiedzy możemy zacząć tworzyć prostą sieć semantyczną faktów na temat drukarki:

Konstrukcja prostej sieci semantycznej opisującej wybrane urządzenie lub aplikację:

```
% Printer information
```

```
% Fakty o drukarce - które udało się wywnioskować z instrukcji
```

```
printer(brother_dcp_t425w).
```

```
color_printer(brother_dcp_t425w).
```

```
inkjet_printer(brother_dcp_t425w).
```

```
wireless_printer(brother_dcp_t425w).
```

```
multifunction_printer(brother_dcp_t425w).
```

```
% Printer capabilities
```

```
% Fakty o możliwościach drukarki - które udało się wywnioskować
```

```
% z dostępnych funkcji oraz samych przycisków
```

```
can_print(brother_dcp_t425w, black_and_white).
```

```
can_print(brother_dcp_t425w, color).
```

```
can_scan(brother_dcp_t425w, black_and_white).
```

```
can_scan(brother_dcp_t425w, color).
```

```
can_copy(brother_dcp_t425w, black_and_white).
```

```
can_copy(brother_dcp_t425w, color).
```

```
can_print_via_wifi(brother_dcp_t425w).
```

```
can_print_via_usb(brother_dcp_t425w).
```

```
% Printer ink cartridge information
```

```
% Fakty o tuszach oraz ich numerach
```

```
ink_cartridge(black_ink_cartridge, btD60BK).
```

```
ink_cartridge(cyan_ink_cartridge, bt5000C).
```

```
ink_cartridge(magenta_ink_cartridge, bt5000M).
```

```
ink_cartridge(yellow_ink_cartridge, bt5000Y).
```

```
% Printer elements
```

```
% Fakty o elementach drukarki
```

```
printer_element(power_cable).
```

```
printer_element(usb_cable).
```

```
printer_element(paper_tray).
```

```
printer_element(paper).
```

```
printer_element(printhead).
```

```
printer_element(black_ink_cartridge).
```

```
printer_element(cyan_ink_cartridge).
```

```
printer_element(yellow_ink_cartridge).
```

```
printer_element(magenta_ink_cartridge).

% Printer panel elements
% Fakty o elementach panelu drukarki
printer_panel_element(power_button).
printer_panel_element(start_color_button).
printer_panel_element(start_mono_button).
printer_panel_element(fast_copy_button).
printer_panel_element(start_mono_button).

% Status Diodes
% Fakty o diodach
printer_panel_element(led_diode_power).
printer_panel_element(led_diode_warinig).
printer_panel_element(ink_led_diode).
printer_panel_element(wifi_diode_button).
```

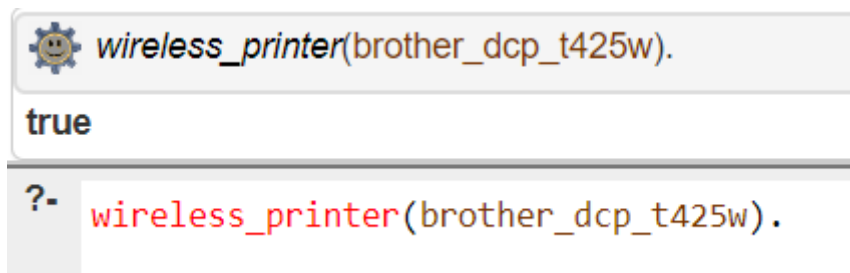
Pokazanie możliwości wnioskowania i działanie unifikacji na kilku dobrze dobranych przykładach zapytań i wnioskowaniu:

Mając te informacje, możemy wykonywać proste zapytania, które będą odpowiadać na pytania typu:

- czy drukarka jest bezprzewodowa?
- w jaki sposób można drukować?
- czy można skanować na drukarce?
- jaki jest numer tuszu czarnego
- jakie elementy posiada drukarka?

Przykładowe zapytania:

?- wireless_printer(brother_dcp_t425w).



?- can_print(brother_dcp_t425w, TRYB).

```
⚙️ can_print(brother_dcp_t425w, TRYB).  
TRYB = black_and_white  
TRYB = color  
?- can_print(brother_dcp_t425w, TRYB).
```

?- *can_scan*(brother_dcp_t425w, color).

```
⚙️ can_scan(brother_dcp_t425w, color).  
true  
?- can_scan(brother_dcp_t425w, color).|
```

?- *ink_cartridge*(black_ink_cartridge, NUMER).

```
⚙️ ink_cartridge(black_ink_cartridge, NUMER).  
NUMER = btD60BK  
?- ink_cartridge(black_ink_cartridge, NUMER).
```

?- *printer_element*(ELEMENT).

```
⚙️ printer_element(ELEMENT).  
ELEMENT = power_cable  
ELEMENT = usb_cable  
ELEMENT = paper_tray  
ELEMENT = paper  
ELEMENT = printhead  
ELEMENT = black_ink_cartridge  
ELEMENT = cyan_ink_cartridge  
ELEMENT = yellow_ink_cartridge  
ELEMENT = magenta_ink_cartridge  
?- printer_element(ELEMENT).|
```



Politechnika
Wrocławska

Autor: Filip Strózik 260377

Prowadzący: Mgr Inż. Michał Karol

?- printer_panel_element(ELEMENT).

```
printer_panel_element(ELEMENT).  
  
ELEMENT = power_button  
ELEMENT = start_color_button  
ELEMENT = start_mono_button  
ELEMENT = fast_copy_button  
ELEMENT = start_mono_button  
ELEMENT = led_diode_power  
ELEMENT = led_diode_warinig  
ELEMENT = ink_led_diode  
ELEMENT = wifi_diode_button  
  
?- printer panel element(ELEMENT).
```

Wszystkie zapytania zwrócą wartość true, ponieważ są to fakty o drukarce. Gdybyśmy mieli fakty o wielu drukarkach, to moglibyśmy poniekąd tworzyć zapytania w celu znalezienia drukarki o określonych właściwościach, np.:

?- wireless_printer(PRINTER), can_print(PRINTER, color), can_scan(PRINTER, color).

```
wireless_printer(PRINTER), can_print(PRINTER, color), can_scan(PRINTER, color).  
  
PRINTER = brother_dcp_t425w  
  
?- wireless_printer(PRINTER), can_print(PRINTER, color), can_scan(PRINTER, color).
```

Zapytanie to zwróciłoby nam wszystkie drukarki, które posiadają możliwość drukowania w kolorze oraz skanowania w kolorze. W naszym przypadku zwróciłoby to tylko jedną drukarkę, ponieważ tylko jedna z nich spełnia te warunki.

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

Po krótkiej lekturze możemy trafić na stronę rozwiązywania problemów, która zawiera takie informacje jak:

Opis przycisków oraz diod:

1. przycisk włączania
2. dioda led zasilania
3. dioda led ostrzegawcza
4. dioda led atramentu
5. przycisk kopiowania w kolorze
6. przycisk kopiowania w czerni
7. przycisk szybkiego kopiowania
8. przycisk / dioda LED Wi-Fi

Również możemy znaleźć informacje o tym jakie błędy mogą wystąpić oraz jak je rozwiązać na podstawie kombinacji świecenia diod.

Dostępne są trzy stany diod:

- wyłączona - off
- włączona - on
- miganie – blink

Obrazkami przedstawione są następujące stany diod oraz ich problemy:

1.

- dioda zasilania - wyłączona
- dioda ostrzegawcza - wyłączona
- dioda tuszu - wyłączona
- problem - drukarka nie jest podłączona do prądu
- element - kabel zasilający

2.

- dioda zasilania - wyłączona
- dioda ostrzegawcza - wyłączona
- dioda tuszu - wyłączona
- problem - drukarka jest wyłączona
- element - przycisk włączania

3.

- dioda zasilania - miganie
- dioda ostrzegawcza - miganie
- dioda tuszu - wyłączona
- problem - zacięcie papieru
- element - papier

4.

- dioda zasilania - włączona
- dioda ostrzegawcza - włączona
- dioda tuszu - wyłączona
- problem - brak papieru
- element - papier

5.

- dioda zasilania - włączona

- dioda ostrzegawcza - miganie
- dioda tuszu - wyłączona
- problem - zły rozmiar papieru
- element – papier

Opracowanie bazy wiedzy w Prologu (bazy klauzul) opisującej wiedzę wyrażoną w tych zdaniach:

Mając zdefiniowane elementy drukarki, dzięki opisom stanów drukarki w instrukcji, możemy zdefiniować, które elementy powodują wystąpienie błędu, a które nie. To przy opisie stanów drukarki za pomocą diod, pozwoli na określenie, które elementy są przyczyną wystąpienia błędu i w jaki sposób można mu zaradzić.

```
% LED states representing error states
% Fakty o stanach diod, które odpowiadają za błędy
% led_state(Error, Power, Warning, Ink)
led_state(not_plugged_in, off, off, off).
led_state(turned_off, off, off, off).
led_state(paper_jam, blink, blink, off).
led_state(no_paper, on, on, off).
led_state(paper_tray_not_inserted, on, on, off).
led_state(incorrect_paper_size, on, blink, off).

% Error results in a specific state
% Fakty o błędach, które powodują wystąpienie konkretnego stanu diod
problem(no_power, not_plugged_in). % drukarka nie jest podłączona do
prądu
problem(off, turned_off). % drukarka wyłączona
problem(dont_print, paper_jam). % zacięcie papieru
problem(dont_print, no_paper). % brak papieru
problem(dont_print, incorrect_paper_size). % nieprawidłowy rozmiar
papieru
problem(dont_print, paper_tray_not_inserted). % nieprawidłowy
rozmiar papieru

print_problem(bad_quality, dirty_printhead). % zła jakość wydruku
print_problem(good_quality, low_cyan_ink). % zła jakość wydruku
print_problem(good_quality, low_magenta_ink). % zła jakość wydruku
print_problem(good_quality, low_yellow_ink). % zła jakość wydruku
print_problem(good_quality, low_black_ink). % zła jakość wydruku

% Element causes a specific Error
% Fakty o elementach, które powodują wystąpienie konkretnego błędu
cause(power_cable, no_power, not_plugged_in). % kabel zasilający nie
jest podłączony
cause(power_button, off, turned_off). % kabel zasilający nie jest
podłączony
cause(paper, dont_print, no_paper). % brak papieru
```



```
cause(paper_tray, dont_print, paper_tray_not_inserted). % taca na
papier nie jest włożona
cause(paper, dont_print, incorrect_paper_size). % papier nie pasuje
do drukarki
cause(paper, dont_print, paper_jam). % papier powoduje zacięcie
papieru
cause(printhead, bad_quality, dirty_printhead). % głowica drukująca
jest zanieczyszczona
cause(cyan_ink_cartridge, good_quality, low_cyan_ink). % brak tuszu
cyan
cause(magenta_ink_cartridge, good_quality, low_magenta_ink). % brak
tuszu magenta
cause(yellow_ink_cartridge, good_quality, low_yellow_ink). % brak
tuszu yellow
cause(black_ink_cartridge, good_quality, low_black_ink). % brak
tuszu black

% Solutions to problems
% Fakty o rozwiązaniach problemów
fix(no_paper, 'insert a paper').
fix(not_plugged_in, 'insert a power cable').
fix(turned_off, 'turn on a printer').
fix(paper_tray_not_inserted, 'insert a paper tray properly').
fix(incorrect_paper_size, 'insert a proper paper size').
fix(paper_jam, 'remove a paper jam').
fix(dirty_printhead, 'clean a printhead').
fix(low_cyan_ink, 'fill up cyan ink cartridge').
fix(low_magenta_ink, 'fill up magenta ink cartridge').
fix(low_yellow_ink, 'fill up yellow ink cartridge').
fix(low_black_ink, 'fill up black ink cartridge').
```

Na tym etapie jesteśmy w stanie zdefiniować predykaty które będą pomagać użytkownikowi w rozwiązywaniu problemów definiowaniu wadliwych elementów oraz akcji, które musi on wykonać aby rozwiązać dany problem. Zauważono, że problemy związane z jakością wydruku nie są sygnalizowane przez żadne z diod. Zatem mamy rozdzielenie na problemy z drukarką które powodują, że drukarka albo jest wyłączona albo jest nie podłączona albo nie drukuje, oraz na takie które powodują złą jakość wydruku i dobrą jakość wydruku z ewentualnym brakiem jakiegoś koloru.

Jest to kluczowe założenie, ponieważ niski poziom jakiegokolwiek z tuszu oraz zanieczyszczona głowica drukująca nadal pozwala na drukowanie, i nie jest to w żaden sposób sygnalizowane, widać to tylko po jakości wydruku.

Predykaty zdefiniowane do wnioskowania i rozwiązywania problemów:

```
define_error_by_led(POWER, WARN, INK, ERROR) :-  
    led_state(ERROR, POWER, WARN, INK).
```

```
define_element(POWER, WARN, INK, ELEMENT) :-  
    define_error_by_led(POWER, WARN, INK, ERROR),  
    problem(STATE, ERROR),  
    cause(ELEMENT, STATE, ERROR),  
    (  
        printer_element(ELEMENT);  
        printer_panel_element(ELEMENT)  
    ).
```

```
define_printing_element(PRINT_QLTY, ELEMENT) :-  
    print_problem(PRINT_QLTY, ERROR),  
    cause(ELEMENT, PRINT_QLTY, ERROR),  
    printer_element(ELEMENT).
```

```
define_solution(ERROR, ACTION) :-  
    fix(ERROR, ACTION).
```

```
troubleshoot(POWER, WARN, INK, ACTION) :-  
    define_error_by_led(POWER, WARN, INK, ERROR),  
    define_solution(ERROR, ACTION).
```

```
troubleshoot_printing_quality(PRINT_QLTY, ACTION) :-  
    print_problem(PRINT_QLTY, ERROR),  
    define_solution(ERROR, ACTION).
```

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

Dzięki tym predykatom możemy określić, które elementy drukarki powodują wystąpienie błędu oraz jakie rozwiązanie jest potrzebne, aby go naprawić. Dodatkowo możemy określić, które elementy drukarki powodują wystąpienie błędu w jakości wydruku.

Przykłady użycia predykatów:

- możemy sprawdzić jaki stan diod odpowiada problemowi:

?- define_error_by_led(on, blink, off, ERROR).



`define_error_by_led(on, blink, off, ERROR).`


ERROR = incorrect_paper_size

?-

`define_error_by_led(on, blink, off, ERROR).`

?- define_error_by_led(POWER, WARN, INK, paper_jam).

```

 define_error_by_led(POWER, WARN, INK, paper_jam).

INK = off,
POWER = WARN, WARN = blink


?- define_error_by_led(POWER, WARN, INK, paper_jam).|

```

- możemy sprawdzić, które elementy powodują wystąpienie błędu:

?- define_element(off, off, off, ELEMENT).

```

 define_element(off, off, off, ELEMENT).


ELEMENT = power_cable
ELEMENT = power_button

?- define_element(off, off, off, ELEMENT).|

```

?- define_element(on, on, off, ELEMENT).

```

 define_element(on, on, off, ELEMENT).


ELEMENT = paper
ELEMENT = paper_tray

?- define_element(on, on, off, ELEMENT).|

```

?- define_element(blink, blink, off, ELEMENT).

```

 define_element(blink, blink, off, ELEMENT).

ELEMENT = paper
false

?- define_element(blink, blink, off, ELEMENT).|

```

Ze względu na rzeczywistość, powyższy wynik jest poprawny, ponieważ stany diod (off, off, off) wskazują na brak prądu lub wyłączenie, które odpowiednio wskazują na przewód i przycisk włączenia. Przewód jest elementem drukarki a przycisk jest elementem panelu drukarki. Występująca tam alternatywa nie jest wymagana by urzeczywistnić realia rozwiązywania problemu drukarki. W przypadku (blink, blink, off) papier jest tylko elementem wewnętrznym, aczkolwiek przez alternatywę będzie sprawdzane czy papier jest elementem panelu drukarki co oczywiście jest fałszem.


?- define_element(POWER, WARN, INK, paper_tray).

 `define_element(POWER, WARN, INK, paper_tray).`

```
INK = off,  
POWER = WARN, WARN = on  
false
```

?- `define_element(POWER, WARN, INK, paper_tray)`.


- możemy zdefiniować, które elementy powodują wystąpienie błędu w jakości wydruku:
?- define_printing_element(bad_quality, ELEMENT).

 `define_printing_element(bad_quality, ELEMENT).`

```
ELEMENT = printhead
```

?- `define_printing_element(bad_quality, ELEMENT)`.

?- define_printing_element(good_quality_but_no_color, ELEMENT).


 `define_printing_element(good_quality_but_no_color, ELEMENT).`

```
ELEMENT = cyan_ink_cartridge  
ELEMENT = magenta_ink_cartridge  
ELEMENT = yellow_ink_cartridge  
ELEMENT = black_ink_cartridge
```

?- `define_printing_element(good_quality_but_no_color, ELEMENT)`.

Jak już wcześniej podkreślono Stany diod led nie określają błędu spowodowanego z jakością wydruku lub brakiem jakiegoś koloru, dlatego został zdefiniowany osobny predykatów do wnioskowania informacji dotyczących tego założenia.


- możemy zdefiniować, jakie czynności są wymagane do naprawy konkretnego błędu:
?- define_solution(incorrect_paper_size, ACTION).

 `define_solution(incorrect_paper_size, ACTION).`

ACTION = 'insert a proper paper size'

?- `define_solution(incorrect_paper_size, ACTION).`


- ?- define_solution(dirty_printhead, ACTION).

 `define_solution(dirty_printhead, ACTION).`

ACTION = 'clean a printhead'

?- `define_solution(dirty_printhead, ACTION).`|


- ?- define_solution(low_black_ink, ACTION).

 `define_solution(low_black_ink, ACTION).`

ACTION = 'fill up black ink cartridge'

?- `define_solution(low_black_ink, ACTION).`|

- możemy wnioskować jaka czynność jest potrzebna podając stany diod:
?- troubleshoot(off, off, off, ACTION).

 `troubleshoot(off, off, off, ACTION).`

ACTION = 'insert a power cable'

ACTION = 'turn on a printer'

?- `troubleshoot(off, off, off, ACTION).`|

- ?- troubleshoot(on, on, off, ACTION).

 `troubleshoot(on, on, off, ACTION).`

ACTION = 'insert a paper'

ACTION = 'insert a paper tray properly'

?- `troubleshoot(on, on, off, ACTION).`

?- troubleshoot(on, blink, off, ACTION).



troubleshoot(on, blink, off, ACTION).

ACTION = 'insert a proper paper size'

?-

troubleshoot(on, blink, off, ACTION).

- możemy wnioskować jaka czynność jest potrzebna określając jakość wydruku:

?- troubleshoot_printing_quality(bad_quality, ACTION).



troubleshoot_printing_quality(bad_quality, ACTION).

ACTION = 'clean a printhead'

?-

troubleshoot_printing_quality(bad_quality, ACTION).

?- troubleshoot_printing_quality(good_quality_but_no_color, ACTION).



troubleshoot_printing_quality(good_quality_but_no_color, ACTION).

ACTION = 'fill up cyan ink cartridge'

ACTION = 'fill up magenta ink cartridge'

ACTION = 'fill up yellow ink cartridge'

ACTION = 'fill up black ink cartridge'

?-

troubleshoot_printing_quality(good_quality_but_no_color, ACTION).

Zdecydowano się na oprogramowanie przypadku użycia w którym użytkownik chce rozwiązać jakiś problem. Program „prowadzi go za rękę” w celu rozwiązania problemu.

```
troubleshoot_script :-  
    write('Is the printer printing?'), nl,  
    read(Answer),  
    (Answer = n ->  
        write('What is the state of the Power diode? (on / blink / off)'), nl,  
        read(Power),  
        write('What is the state of the Warn diode? (on / blink / off)'), nl,  
        read(Warn),  
        write('What is the state of the Ink diode? (on / blink / off)'), nl,  
        read(Ink),  
        troubleshoot(Power, Warn, Ink, ACTION),  
        write(ACTION), nl  
    ;  
    Answer = y ->  
        write('What is the quality of printing? (bad_quality /  
good_quality_but_no_color)'), nl,  
        read(Quality),  
        troubleshoot_printing_quality(Quality, ACTION),  
        write(ACTION), nl  
    ).
```



troubleshoot_script.

Is the printer printing?

n

What is the state of the Power diode? (on / blink / off)

on

What is the state of the Warn diode? (on / blink / off)

on

What is the state of the Ink diode? (on / blink / off)

off

insert a paper

true

insert a paper tray properly

true



troubleshoot_script.

Is the printer printing?

n

What is the state of the Power diode? (on / blink / off)

off

What is the state of the Warn diode? (on / blink / off)

off

What is the state of the Ink diode? (on / blink / off)

off

insert a power cable

true

turn on a printer

true



troubleshoot_script.

Is the printer printing?

n

What is the state of the Power diode? (on / blink / off)

blink

What is the state of the Warn diode? (on / blink / off)

blink

What is the state of the Ink diode? (on / blink / off)

off

remove a paper jam

true

false



troubleshoot_script.

Is the printer printing?

y

What is the quality of printing? (bad_quality / good_quality_but_no_color)

good_quality_but_no_color

fill up cyan ink cartridge

true

fill up magenta ink cartridge

fill up yellow ink cartridge

fill up black ink cartridge

true

true

true



troubleshoot_script.

Is the printer printing?

y

What is the quality of printing? (bad_quality / good_quality_but_no_color)

bad_quality

clean a printhead

true

Wnioski

Korzystając z reguł diagnostycznych, zasad postępowania oraz możliwości rozwiązania potencjalnych problemów, które wydobyliśmy z rozdziału poświęconego rozwiązywaniu problemów, przekształciliśmy zdobytą wiedzę w dodatkowe klauzule i predykaty. Wykorzystując opracowaną bazę wiedzy, byliśmy w stanie rozwiązać każdy problem przedstawiony w instrukcji obsługi drukarki.

W trakcie przeprowadzania ćwiczenia, dążyliśmy do oddania jak najbardziej realistycznego stanu drukarki, opierając się na faktach i informacjach dostępnych w instrukcji. Naszym celem było skuteczne wykorzystanie faktycznych danych o drukarce w celu rozwiązania problemów i osiągnięcia zamierzonych rezultatów.

Podsumowując, przeprowadzone ćwiczenie umożliwiło nam praktyczne zastosowanie logiki predykatów pierwszego rzędu i zasady rezolucji w kontekście rozwiązywania problemów związanych z konkretnym urządzeniem. Dzięki wykorzystaniu informacji o drukarce Brother DCP-T425W zawartych w instrukcji obsługi, byliśmy w stanie skutecznie rozwiązać każdy przedstawiony problem, co świadczy o powodzeniu wykonanego zadania. Mając odpowiednie fakty oraz reguły dotyczące jakiegokolwiek urządzenia czy systemu, możemy automatyzować proces diagnostyki i rozwiązywania problemów. Dla użytkownika byłoby to szalenie łatwe by tylko podawać stan urządzenia, po czym dostać listę kroków do wykonania by rozwiązać problem.