

**Sprawozdanie z ćwiczeń laboratoryjnych z przedmiotu
„Podstawy Sztucznej Inteligencji” AGH 2020/2021
z zajęć laboratoryjnych „Eksploracja procesów”**

<https://upel2.cel.agh.edu.pl/weaiib/course/view.php?id=270>



Imię i nazwisko:

Joanna Hankus

Maciej Żądło

Sprawozdanie może być realizowane w parach, natomiast w sprawozdaniu należy podać wtedy imiona i nazwiska osób z pary oraz każda osoba powinna wysłać sprawozdanie przez system UPEL osobno.

Grupa: 2a

Na podstawie instrukcji do ćwiczeń laboratoryjnych na Google Colab:

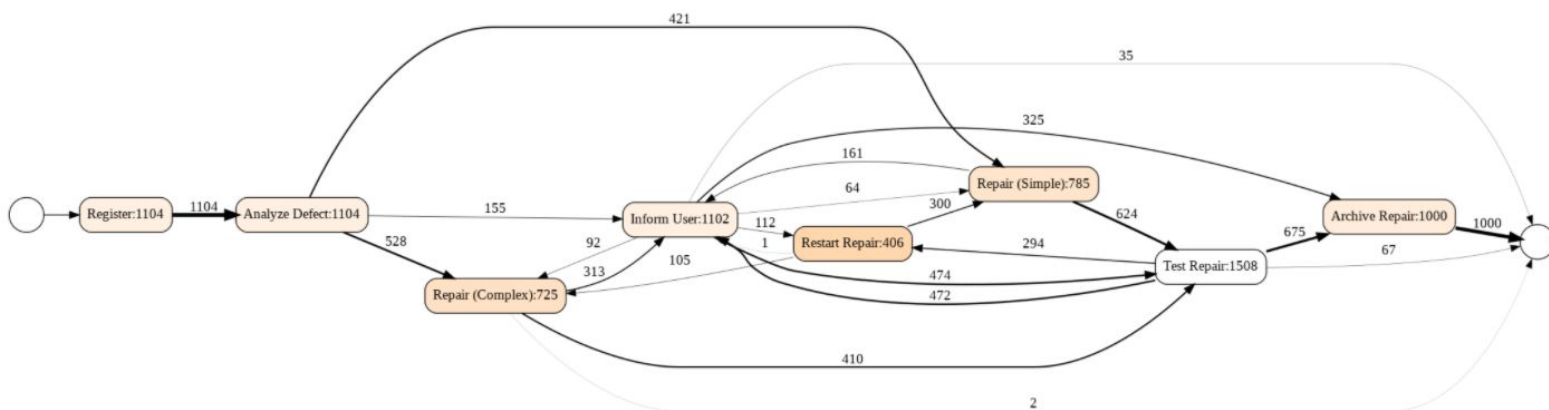
https://colab.research.google.com/drive/1DWY6a-KsZlYm_pwIn0OuCP7jF9_3UEDX?usp=sharing

odpowiedz na następujące pytania:

1. **[1 pkt.]** Ilu testerów oraz solverów jest zaangażowanych w proces?
6 Testerów i 6 Solverów, 1510 wystąpień solverów oraz 2612 testerów.
2. **[1 pkt.]** Które zadanie zajmuje najwięcej czasu?
O indexie 112 - Repair Complex
3. **[1 pkt.]** Ile w analizowanym logu jest wszystkich zdarzeń (wpisów w logu), ile śladów (przypadków), a także ile wariantów różnych śladów można w nim wyróżnić?
*Wszystkich zapisów jest 8838
Wariantów jest 60
oraz 1104 przypadki*
4. **[1 pkt.]** Jak wygląda najczęstszy ślad (najpopularniejszy wariant)?
Register;Analyze Defect;Repair (Complex);Inform User;Test Repair;Archive Repair;End;
5. **[3 pkt.]** Jakie podstawowe informacje dotyczące zdarzenia są konieczne do stworzenia z danych dziennika zdarzeń, który będzie odpowiedni do celu odkrywania modelu procesu biznesowego?
Jeżeli chcemy stworzyć model procesu biznesowego, to będziemy potrzebowali informacji o ścieżkach (czyli możliwościach ich ułożenia oraz ich liczebności). Dzięki temu będziemy mogli analizować zdarzenie.
6. **[3 pkt.]** Na podstawie obserwacji otrzymanego modelu, jakie wnioski można wyciągnąć z odkrytego procesu naprawy telefonów?
 - *Na początku następuje zawsze Rejestracja i Analiza usterki*
 - *Najczęściej podejmowanymi działaniami są Test Repair i Repair(Simple)*
7. **[2 pkt.]** Zmodyfikuj etykiety zadań w modelu, tak by dla każdego zadania oprócz nazwy wyświetlana była także liczba wystąpień odpowiadającego mu zdarzenia w logu. Zamieść odpowiedni fragment kodu oraz zrzut ekranu.

**Sprawozdanie z ćwiczeń laboratoryjnych z przedmiotu
„Podstawy Sztucznej Inteligencji” AGH 2020/2021
z zajęć laboratoryjnych „Eksploracja procesów”**

<https://upel2.cel.agh.edu.pl/weaiib/course/view.php?id=270>



```
G = pgv.AGraph(strict=False, directed=True)
G.graph_attr['rankdir'] = 'LR'
G.node_attr['shape'] = 'Mrecord'
for event, successors in w_net.items():
    value = ev_counter[event]
    color = int(float(color_max-value)/float(color_max-color_min)*100.00)
    my_color = "#ff9933"+str(hex(color))[2:]
    info = str(event) + ":" + str(value)
    G.add_node(event, style="rounded, filled", fillcolor=my_color, label = info)
    for pr, cnt in successors.items(): # preceding event, count
        G.add_edge(event, pr, penwidth=4*cnt/(trace_max-trace_min)+0.1, label=cnt)

ev_source = set(w_net.keys())
ev_target = reduce(lambda x,y: x | y, [w.keys() for w in w_net.values()])
ev_start_set = ev_source - ev_target # start events
ev_end_set = ev_target - ev_source # end events

for ev_end in ev_end_set:
    end = G.get_node(ev_end)
    end.attr['shape'] = 'circle'
    end.attr['label'] = ''

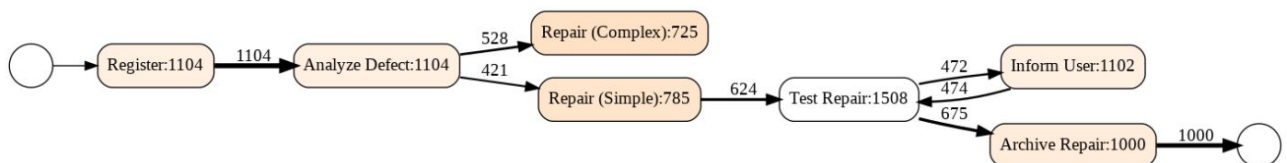
G.add_node("start", shape="circle", label="")
for ev_start in ev_start_set:
    G.add_edge("start", ev_start)

G.draw('simple_heuristic_net_with_events.png', prog='dot')
display(Image('simple_heuristic_net_with_events.png'))
```

8. [5 pkt.] Dodaj opcję filtrowania (po zdarzeniach lub przepływach) w zależności od określonych progów, aby pokazać lub ukryć zadania lub przepływy zgodnie z wybranym progiem. Zaprezentuj modele dla wybranych progów np. 420 dla przepływów, 700 dla zdarzeń (osobno i razem). Zamieść odpowiedni fragment kodu oraz zrzut ekranu.



```
for event, sucesors in w_net.items():
    value = ev_counter[event]
    if value > 700:
        color = int(float(color_max-value)/float(color_max-color_min)*100.00)
        my_color = "#ff9933"+str(hex(color))[2:]
        info = str(event) + ":" + str(value)
        G.add_node(event, style="rounded,filled", fillcolor=my_color, label = info)
        for pr, cnt in sucesors.items(): # preceeding event, count
            if cnt > 420:
                G.add_edge(event, pr, penwidth=4*cnt/(trace_max-trace_min)+0.1, label=cnt)
```



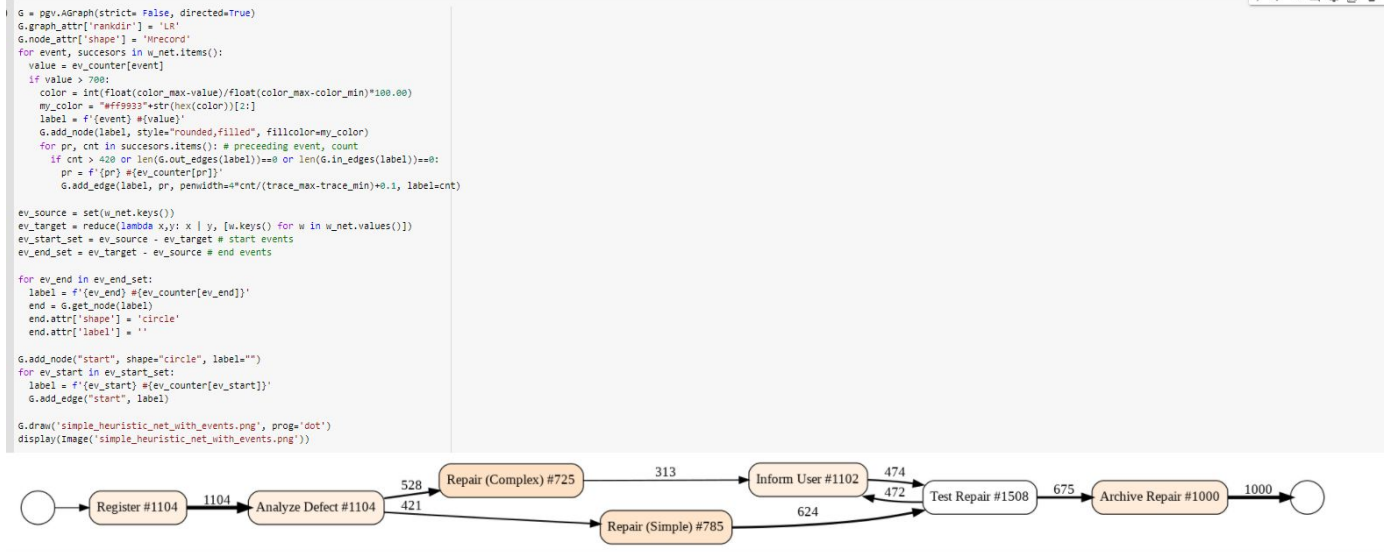
9. [7 pkt.] Przetestuj różne progi, aby określić, czy model wygląda prawidłowo, w szczególności, czy jakieś zdania nie zostają odczepione od modelu lub nie są prawidłowo połączone z modelem, np. jeśli wcześniej zadanie występowało pomiędzy innymi zadaniami nie należy filtrować wszystkich przepływów, nawet jeśli są poniżej progu. Zmodyfikuj progowanie, tak aby w takim wypadku zachować najlepszy przepływ, aby zadanie było połączone z pozostałymi co najmniej jednym wchodzącym i co najmniej jednym wychodzącym przepływem. Pokaż działanie swoich modyfikacji na zrzutach ekranu, jak również wklej do sprawozdania odpowiednie fragmenty kodu.

**Sprawozdanie z ćwiczeń laboratoryjnych z przedmiotu
„Podstawy Sztucznej Inteligencji” AGH 2020/2021
z zajęć laboratoryjnych „Eksploracja procesów”**

<https://upe12.cel.agh.edu.pl/weaiib/course/view.php?id=270>



Warunek odnośnie istnienia z wychodzącym/wchodzącym przepływem:



Sprawozdanie z ćwiczeń laboratoryjnych z przedmiotu „Podstawy Sztucznej Inteligencji” AGH 2020/2021 z zajęć laboratoryjnych „Eksploracja procesów”

<https://upe12.cel.agh.edu.pl/weaiib/course/view.php?id=270>



Przykład dla którego zwiększymy próg i zostawimy warunek odnośnie przepływu (w takim wypadku graf zostanie po prostu “rozerwany”):

