ADITHYA RAGHU GANESH
200314659

CSC520 Fall 2019 Assignment 4
Due November 5th at 11:59pm

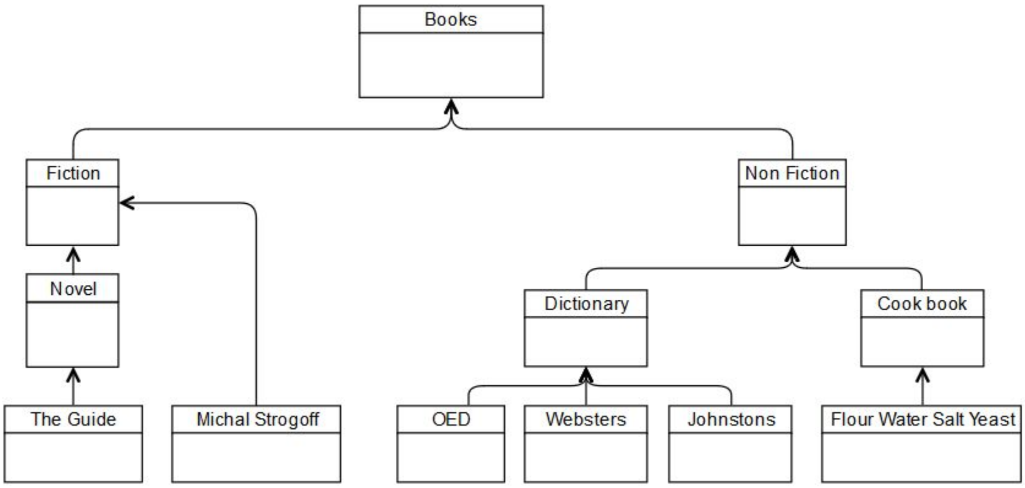
Question 1 (20 pts)

Consider the following facts:

All books are either fiction or nonfiction. Dictionaries are nonfiction, so are cookbooks. The OED is a dictionary as are Webster's and Johnston's dictionaries. "Michael Strogoff" is fifiction. "The Guide" is a Novel. And all novels are fiction. "Flour Water Salt Yeast" is a cookbook.

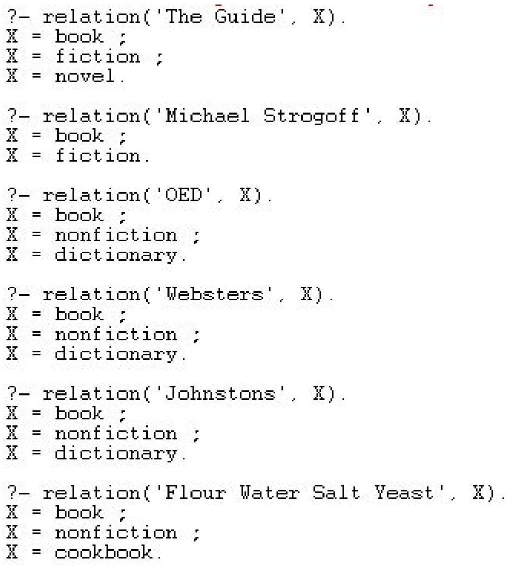
A. (5 pts) Draw a frame representation of these facts with appropriate labels for the nodes and edges.Include this in your report.

Answer:



B. (5 pts) Using a consistent top level syntax as discussed in class along with other necessary predicates write a Prolog representation of the taxonomy along with one or more rules to infer that each book described is a novel, a dictionary, fiction, or nonfiction etc. as appropriate and that they are all books. Submit this as a file called q1b.pl

Answer:

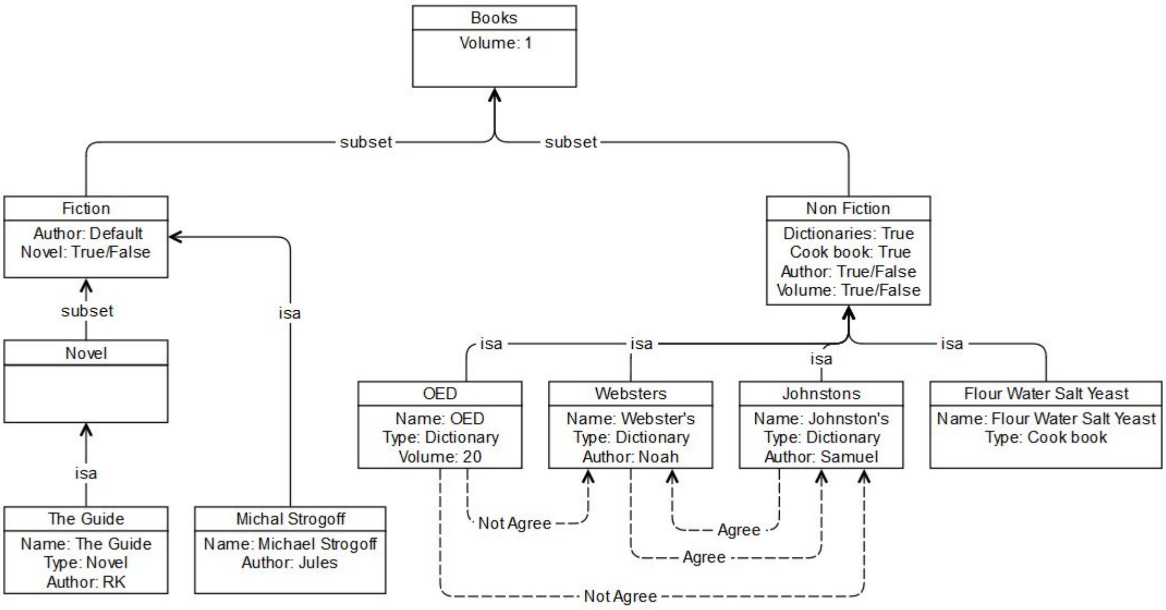


```
?- relation('The Guide', X).
X = book ;
X = fiction ;
X = novel.

?- relation('Michael Strogoff', X).
X = book ;
X = fiction.

?- relation('OED', X).
X = book ;
X = nonfiction ;
X = dictionary.

?- relation('Websters', X).
X = book ;
X = nonfiction ;
X = dictionary.

?- relation('Johnstons', X).
X = book ;
X = nonfiction ;
X = dictionary.

?- relation('Flour Water Salt Yeast', X).
X = book ;
X = nonfiction ;
X = cookbook.
```

ADITHYA RAGHU GANESH
200314659

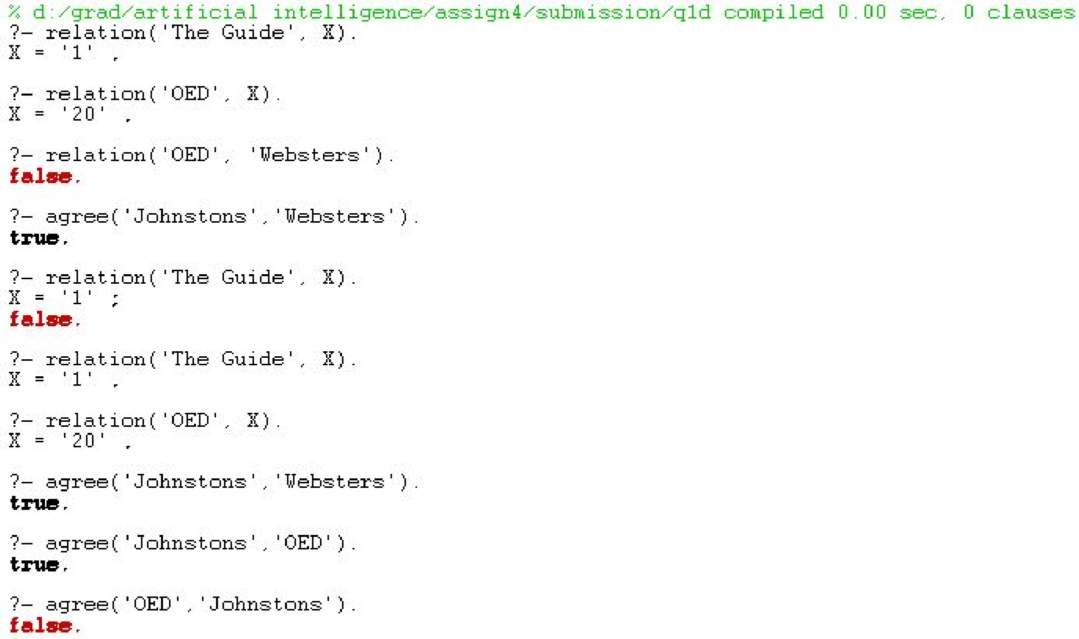
C. (5 pts) Now expand to a frame representation that also incorporates the following information: All books have editors. All fiction has authors. Webster's and Johnston's dictionaries have authors (Noah and Samuel respectively). The OED does not. The OED has 20 volumes. Most other books have 1. The author of "Michael Strogoff" is Jules while the author of "The Guide" is RK. Also show that Webster's and Johnston's dictionaries agree with each other. While Johnston's agrees with the OED but not vice-versa.

Answer:



D. (5 pts) Enhance your code to include the above knowledge and to allow for the inference that the number of volumes of "The Guide" is 1 and that Johnston's dictionary agrees with everyone but that the OED does not agree with anyone. Submit this as a file called q1d.pl

Answer:



```
% d:/grad/artificial intelligence/assign4/submission/q1d compiled 0.00 sec, 0 clauses
?- relation('The Guide', X).
X = '1' .

?- relation('OED', X).
X = '20' .

?- relation('OED', 'Websters').
false.

?- agree('Johnstons','Websters').
true.

?- relation('The Guide', X).
X = '1' ;
false.

?- relation('The Guide', X).
X = '1' .

?- relation('OED', X).
X = '20' .

?- agree('Johnstons','Websters').
true.

?- agree('Johnstons','OED').
true.

?- agree('OED','Johnstons').
false.
```

ADITHYA RAGHU GANESH
200314659

Question 2 (20 pts)

For this question consider the following problem description:

You need to install the new engine in your car, as part of that you need a fresh set of spark plugs and a fresh belt, neither of which you have. However your garage is clean. To change the plugs you need to have no plugs and a clean garage to work in. To change the belt you need no belt and a clean garage. Changing the belt or changing the plugs will get you a new belt or new plugs but doing either one will mess up your garage. When the garage is dirty you can always clean it. At the end of the day you want an engine and a clean garage.

A. (5 pts) Provide a PDDL representation of the above problem specification.

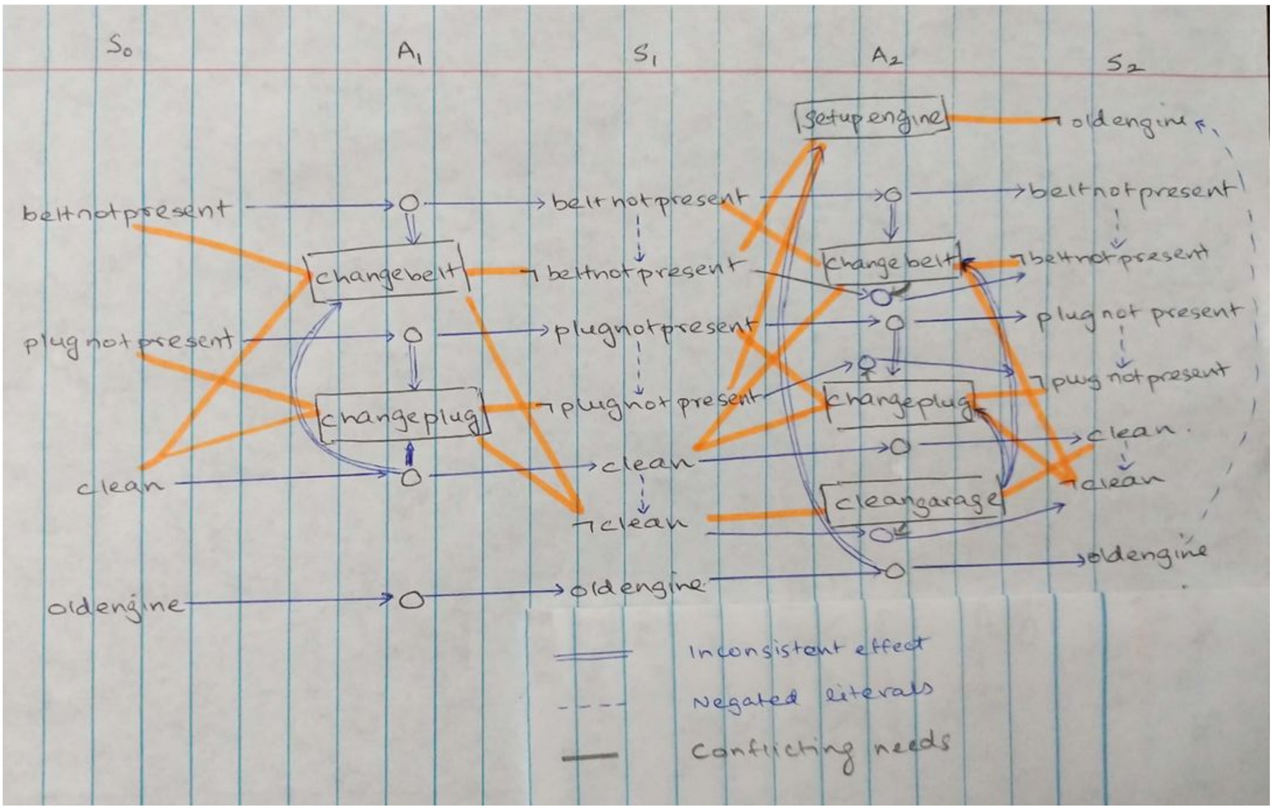
Answer:

```
(define (domain engineinstall)
(:types car - object
garage - object)

(:predicates (beltnotpresent ?x)
        (plugnotpresent ?x)
        (clean ?y)
        (dirty ?y)
        (newbelt ?x)
        (newplug ?x)
        (newengine ?x)
        (oldengine ?x))

(:init (and (oldengine ?x)
(plugnotpresen ?xt)
(beltnotpresent ?x)
(clean ?y)))

(:goal (and (newengine ?x)
(newplug ?x)
(newbelt ?x)
(clean ?y)))

(action: changebelt
:parameters (?x - car ?y - garage)
:precondition (and (beltnotpresent ?x)
        (clean ?y))
:effect (and (newbelt ?x)
(not (clean ?y))))

(action: changeplug
:parameters (?x - car ?y - garage)
:precondition (and (plugnotpresent ?x)
        (clean ?y))
:effect (and (newplug ?x)
(not (clean ?y))))
(action: cleangarage
:parameters (?y - garage)
:precondition (and not (clean ?y))
:effect (and (clean ?y)
(not (dirty ?y))))

(action: setupengine
:parameters (?x - car ?y - garage)
:precondition (and (newplug ?x)
          (newbelt ?x)
          (clean ?y)
          (oldengine ?x))
:effect (and (newengine ?x)
(not (oldengine ?x)))))
```

B. (15 pts) Apply the Graphplan algorithm to solve this problem showing the complete steps, mutex links, and the final plan. All levels, actions, variables and mutex links must be clearly labeled and their types shown. Add your graph and the plan to your final report.

Answer:

Key : orange lines define the possible paths



ADITHYA RAGHU GANESH
200314659

Question 3 (60 pts)
For this question you must implement and evaluate the critical path scheduling method in Prolog for arbitrary scheduling graphs. Your code must be designed to work with a database of tasks, prerequisites, and durations of the following form:
duration(a, 10).
prerequisite(b, a).
Where the first states the duration of task a is 10 units and the second that task is a necessary prerequisite for task b. You will be given a pair of graph files which you can use to test your code. You are encouraged to develop your own files for testing as your code will be tested on novel databases. Your final code must implement the following predicates for testing without using the if-then construct:
criticalPath(<task>, <path>).
earlyFinish(<task>, <time>).
lateStart(<task>, <time>).
maxSlack(<task>, <time>).

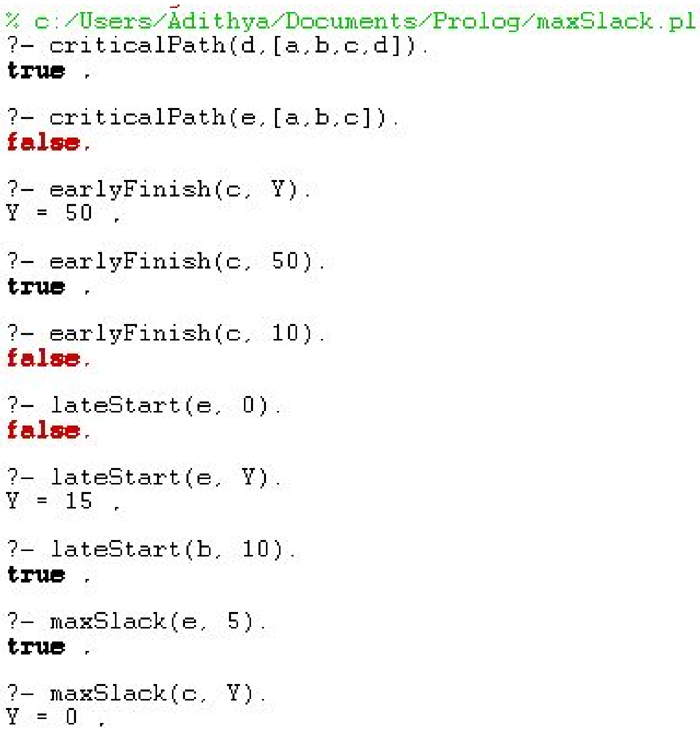
Answer:
Sample A,B values:

Node(Duration)	Critical Path	Early Finish	Late Start	Maximum Slack
a(10)	a	10	0	0
b(20)	a, b	30	10	0
c(20)	a, b, c	50	30	0
d(10)	a, b, c, d	60	50	0
e(35)	a, e	45	15	5

Node(Duration)	Critical Path	Early Finish	Late Start	Maximum Slack
a(10)	a	10	0	0
b(5)	a, b	15	40	30
c(5)	a, b, c	20	45	30
d(9)	a, b, c, d	29	50	30
e(21)	a, b, c, e	41	50	30
f(16)	a, b, c, d. h, f	57	71	30
g(21)	a, l, m, n, j, k, g	108	87	0
h(12)	a, b, c, d, h	41	59	30
i(10)	a, i	20	17	7
j(25)	a, l, m, n, j	52	27	0
k(35)	a, l, m, n, j, k	87	52	0
l(8)	a, l	18	10	0
m(4)	a, l, m	22	18	0
n(5)	a, m, l, n	27	27	0

ADITHYA RAGHU GANESH
200314659

Sample output:



```
% c:/Users/Adithya/Documents/Prolog/maxSlack.pl
?- criticalPath(d,[a,b,c,d]).
true .

?- criticalPath(e,[a,b,c]).
false.

?- earlyFinish(c, Y).
Y = 50 .

?- earlyFinish(c, 50).
true .

?- earlyFinish(c, 10).
false.

?- lateStart(e, 0).
false.

?- lateStart(e, Y).
Y = 15 .

?- lateStart(b, 10).
true .

?- maxSlack(e, 5).
true .

?- maxSlack(c, Y).
Y = 0 .
```