Module 5 Graded Quiz

Due Mar 27 at 11:59pm **Points** 10 **Questions** 10
Available after Mar 13 at 11:59pm
Time Limit 300 Minutes **Allowed Attempts** 3

Take the Quiz Again

Attempt History

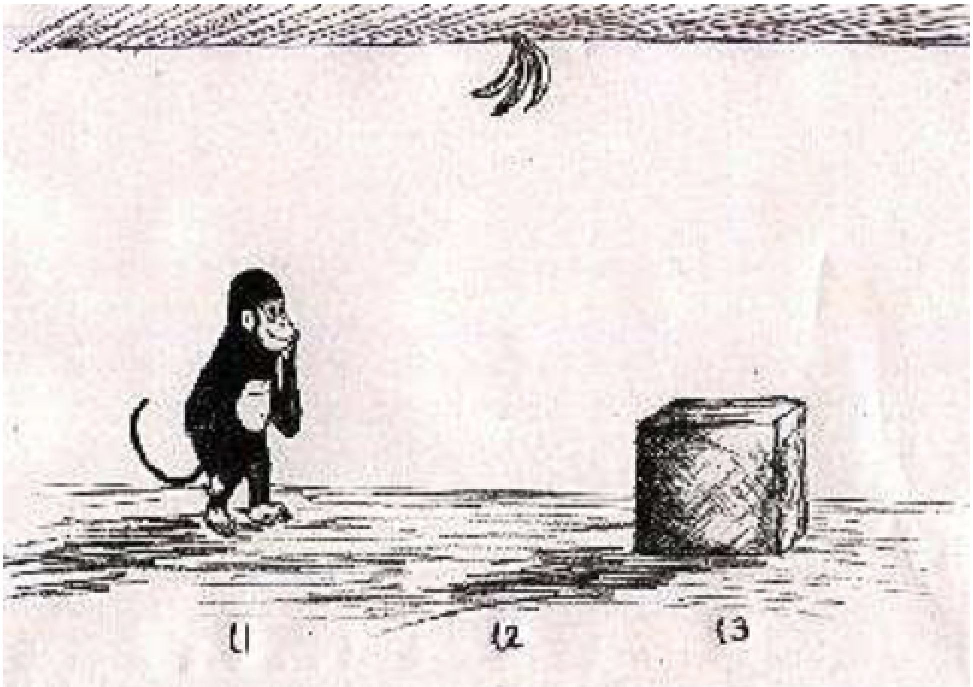
	Attempt	Time	Score
LATEST	Attempt 1	7 minutes	10 out of 10

Score for this attempt: **10** out of 10
Submitted Mar 22 at 4:59pm
This attempt took 7 minutes.

Question 1

1 / 1 pts

Considering "monkey grasp bananas" problem in the following image, which option is a fluent?



Monkey moves the box.

The jump of the monkey.

Monkey grasps the bananas.

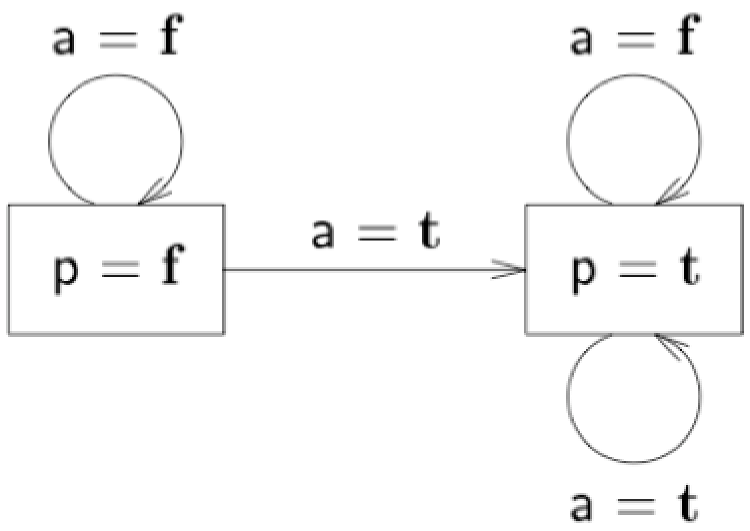
Correct!

The location of the box.

Question 2

1 / 1 pts

Consider the following image, which option represents “actions are exogenous”?



{p(t,1)} :- p(t,0).

1{p(t,0); p(f,0)}1.

p(t,1) :- a(0).

Correct!

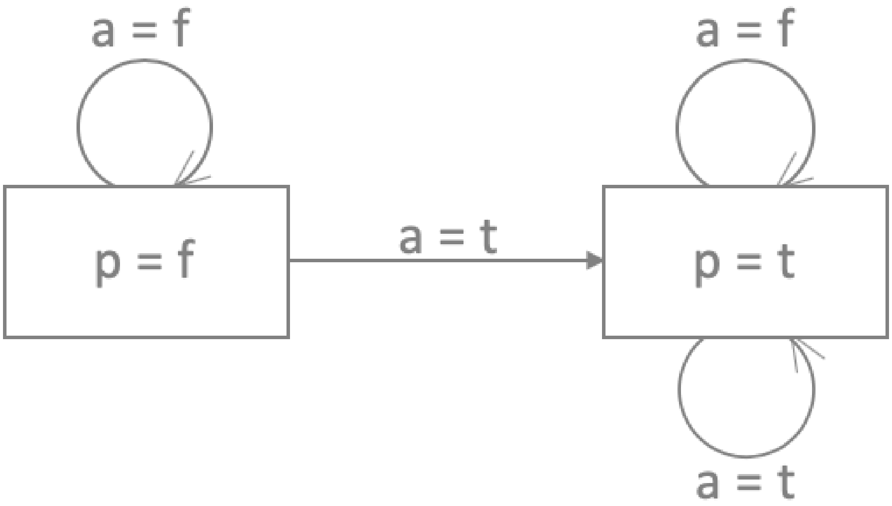
{a(0)}.

Question 3

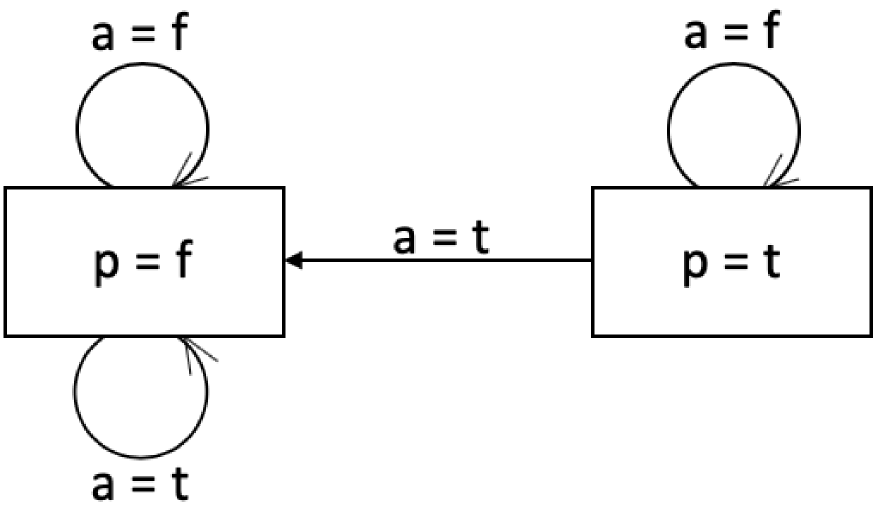
1 / 1 pts

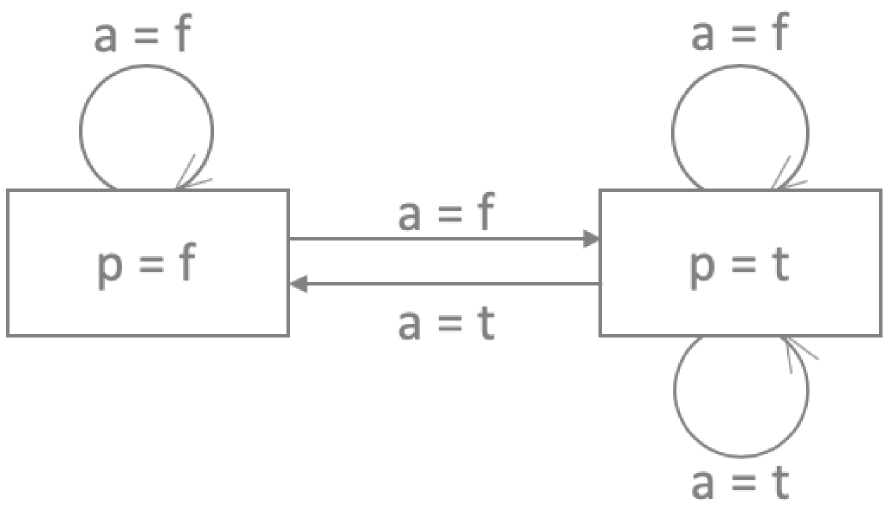
Choose the transition system described by the following ASP program. Note that p denotes a fluent symbol, and a denotes an action symbol. (Hint: the 2nd rule is different from that in the slides.)

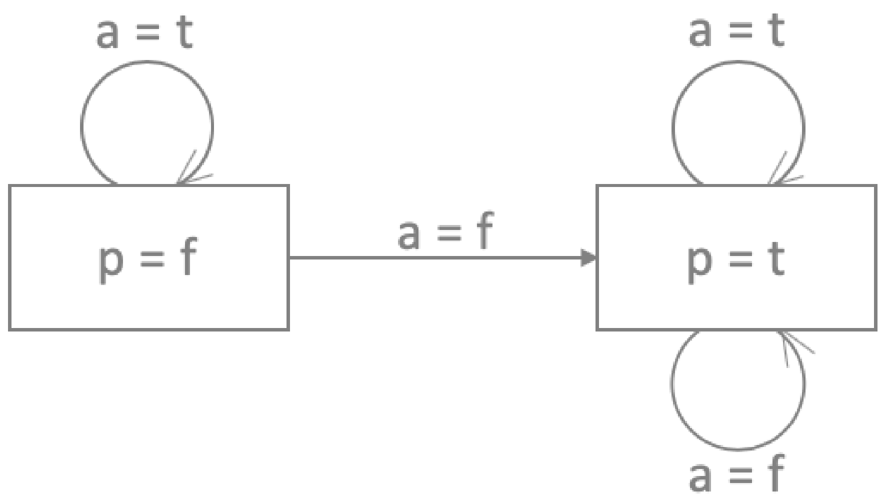
```
boolean(t;f).
p(f, T+1) :- a(T), T=0..m-1.
1{p(B,0): boolean(B)}1.
:- T=1..m, not 1{p(B, T): boolean(B)}1.
{a(T)} :- T=0..m-1.
{p(B, T+1)} :- p(B, T), T=0..m-1.
```



Correct!







Question 4

1 / 1 pts

Consider the same clingo program below that you have seen in question 3. How many stable models are there when m equals to 1?

```
boolean(t;f).
p(f, T+1) :- a(T), T=0..m-1.
1{p(B,0): boolean(B)}1.
:- T=1..m, not 1{p(B, T): boolean(B)}1.
{a(T)} :- T=0..m-1.
{p(B, T+1)} :- p(B, T), T=0..m-1.
```

Correct!

- 4
- 6
- 2
- 8

Question 5

1 / 1 pts

Consider the following clingo program for “Monkey and Bananas” problem.

```
%% sort and object declaration

boolean(t;f).
object(monkey;bananas;box).
location(l1;l2;l3).

%% state constraints

loc(bananas,L,T) :- hasBananas(t,T), loc(monkey,L,T).
loc(monkey,L,T) :- onBox(t,T), loc(box,L,T).

%% effect and preconditions of actions

% walk
loc(monkey,L,T+1) :- walk(L,T).
:- walk(L,T), loc(monkey,L,T).
:- walk(L,T), onBox(t,T).

% pushBox
loc(box,L,T+1) :- pushBox(L,T).
loc(monkey,L,T+1) :- pushBox(L,T).
:- pushBox(L,T), loc(monkey,L,T).
:- pushBox(L,T), onBox(t,T).
:- pushBox(L,T), loc(monkey,L1,T), loc(box,L2,T), L1 != L2.

% climbOn
onBox(t,T+1) :- climbOn(T).
:- climbOn(T), onBox(t,T).
:- climbOn(T), loc(monkey,L1,T), loc(box,L2,T), L1 != L2.
```

```
% climbOff
onBox(f,T+1) :- climbOff(T).
:- climbOff(T), onBox(f,T).

% graspBananas
hasBananas(t,T+1) :- graspBananas(T).
:- graspBananas(T), hasBananas(t,T).
:- graspBananas(T), onBox(f,T).
:- graspBananas(T), loc(monkey,L1,T), loc(bananas,L2,T), L1 != L2.

% disallow concurrent actions
:- walk(L,T), pushBox(L,T).
:- walk(L,T), climbOn(T).
:- pushBox(L,T), climbOn(T).
:- climbOff(T), graspBananas(T).

%% domain independent axioms

% fluents are initially exogenous
1{hasBananas(BB,0):boolean(BB)}1.
1{onBox(BB,0):boolean(BB)}1.
1{loc(O,LL,0):location(LL)}1 :- object(O).

% uniqueness and existence of fluent values
:- not 1{loc(O,LL,T)}1, object(O), T = 1..m.
:- not 1{onBox(BB,T)}1, T = 1..m.
:- not 1{hasBananas(BB,T)}1, T = 1..m.

% actions are exogenous
{walk(L,T)} :- location(L), T = 0..m-1.
{pushBox(L,T)} :- location(L), T = 0..m-1.
{climbOn(T)} :- T = 0..m-1.
{climbOff(T)} :- T = 0..m-1.
{graspBananas(T)} :- T = 0..m-1.

% commonsense law of inertia
{hasBananas(B,T+1)} :- hasBananas(B,T), T=0..m-1.
{onBox(B,T+1)} :- onBox(B,T), T=0..m-1.
{loc(O,L,T+1)} :- loc(O,L,T), T=0..m-1.
```

Which option is a set of rules that makes the following rule redundant?

```
:- walk(L,T), pushBox(L1,T), L!=L1.
```

Correct!

loc(monkey,L,T+1) :- walk(L,T).

loc(monkey,L,T+1) :- pushBox(L,T).

:- not 1{loc(O,LL,T)}1, object(O), T = 1..m.

---

loc(monkey,L,T+1) :- walk(L,T).

:- walk(L,T), loc(monkey,L,T).

loc(monkey,L,T+1) :- pushBox(L,T).

:- pushBox(L,T), loc(monkey,L,T).

:- walk(L,T), onBox(t,T).

:- pushBox(L,T), onBox(t,T).

:- walk(L,T), loc(monkey,L,T).

:- pushBox(L,T), loc(monkey,L,T).

:- not 1{loc(O,LL,T)}1, object(O), T = 1..m.

Question 6

1 / 1 pts

Consider the clingo program for "Monkey and Bananas" problem in question 5. Suppose monkey and bananas are in location L1, and the box is in location L2. If you want to use clingo to generate a plan for the monkey to get the bananas, which option is the set of rules you need to add?

:- not loc(monkey, l1, 0).

:- not loc(bananas, l1, 0).

:- not loc(box, l2, 0).

:- not hasBananas(f, 0).

:- not hasBananas(t, 1).

loc(monkey, l1, 0).

loc(bananas, l1, 0).

loc(box, l2, 0).

hasBananas(f, 0; t, m).

:- not loc(monkey, l1, 0; bananas, l1, 0; box, l2, 0).

hasBananas(f, 0).

hasBananas(t, m).

Correct!

:- not loc(monkey, l1, 0; bananas, l1, 0; box, l2, 0).

:- not hasBananas(f, 0; t, m).

Question 7

1 / 1 pts

Consider the following clingo program for “Blocks World” problem.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% sort and object declaration
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% every block is a location
location(B) :- block(B).

% the table is a location
location(table).

%%%%%%%%%%%%%%%%%%%%%%%%%%
% state description
%%%%%%%%%%%%%%%%%%%%%%%%%%

% two blocks can't be on the same block at the same time
:- 2{on(BB,B,T)}, block(B), T = 0..m.


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% effect and preconditions of action
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% effect of moving a block
on(B,L,T+1) :- move(B,L,T).

% concurrent actions are limited by num of grippers
:- not {move(BB,LL,T)} grippers, T = 0..m-1.

% a block can be moved only when it is clear
:- move(B,L,T), on(B1,B,T).

% a block can't be moved onto a block that is being moved also
:- move(B,B1,T), move(B1,L,T).


%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% domain independent axioms
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% fluents are initially exogenous
1{on(B,LL,0):location(LL)}1 :- block(B).

% uniqueness and existence of value constraints
:- not 1{on(B,LL,T)}1, block(B), T=1..m.

% actions are exogenous
{move(B,L,T)} :- block(B), location(L), T = 0..m-1.

% commonsense law of inertia
{on(B,L,T+1)} :- on(B,L,T), T < m.
```

Suppose we have two grippers G1 and G2. According to the ASP formalization, when A, B are on the table and C is on B, can G1 move C onto the table at the same time G2 moves A onto B?

Correct!

- (selected) Yes
- No

Question 8

1 / 1 pts

Consider the following clingo program discussed in the module "Expressive Possibilities".

```
boolean(t;f).

% sorts and object declarations
object(jack;car).
location(home;work).

% effect and precondition of go
loc(jack,L,T+1) :- go(L,T).
{loc(car,L,T+1)} :- go(L,T), loc(car,L1,T), loc(jack,L1,T), T=0..m-1.
:- go(L,T), loc(jack,L,T).

% fluents are exogenous initially
1{loc(O,LL,0):location(LL)}1 :- object(O).

% uniqueness and existence of fluent values
:- not 1{loc(O,LL,T)}1, object(O), T=1..m.

% actions are exogenous
{go(L,T)} :- location(L), T=0..m-1.

% fluents are inertial
{loc(O,L,T+1)} :- loc(O,L,T), T=0..m-1.
```

Suppose Jack and his car are at home at time 0 and he goes to work at time 0. How many stable models are there that following this condition when m equals to 1?

- 0
- 1
- 4

Correct!

- 2

Question 9

1 / 1 pts

Consider the following clingo program for “lifting” problem.

```
% sort and object declarations
boolean(t;f).
end(leftEnd;rightEnd).
height(low;high).

% state condition
onTable(f,T) :- level(leftEnd,H,T), level(rightEnd,H1,T), H!=H1.

% effect and precondition of lift
level(E,high,T+1) :- lift(E,T).
:- lift(E,T), level(E,high,T).

% fluents are exogenous initially
1{level(E,HH,0): height(HH)}1 :- end(E).
1{onTable(BB,0): boolean(BB)}1.

% uniqueness and existence of fluent values
:- not 1{level(E,HH,T)}1, end(E), T=1..m.
:- not 1{onTable(BB,T)}1, T=1..m.

% actions are exogenous
{lift(E,T)} :- end(E), T=0..m-1.

% fluents are inertial
{level(E,H,T+1)} :- level(E,H,T), T=0..m-1.
{onTable(B,T+1)} :- onTable(B,T), T=0..m-1.
```

When m equals to 1, how many stable models of this program satisfy the following scenario?

```
% scenario
onTable(t,0).
level(E,low,0) :- end(E).
:- not onTable(t,m).
:- level(E,low,m).
```

- 2
- 0
- 4

Correct!

- 1

Question 10

1 / 1 pts

Which option is correct about action description languages?

Correct!

- (selected) They are used to describe the preconditions and effects of actions.
- They always map to answer set programs.
- The fluents in an action language have values, and the values are either t or f.
- The “commonsense law of inertia” describes the effect of actions.

Quiz Score: **10** out of 10

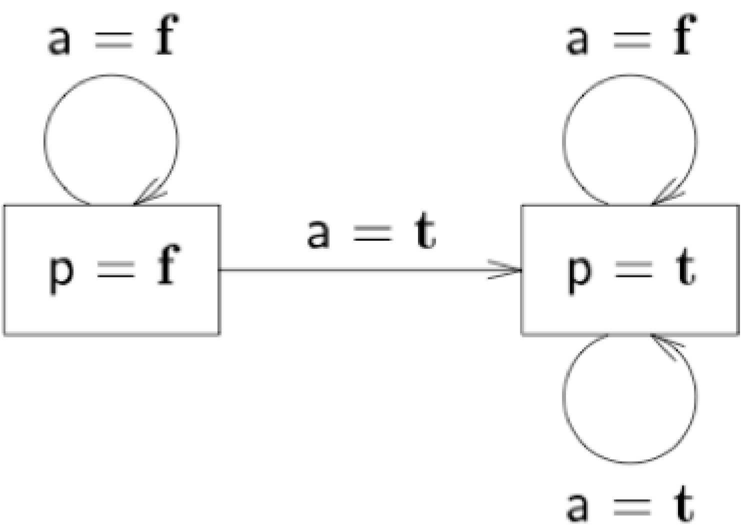
Module 5 Practice Quiz Results for Anil Kumar

Submitted May 3 at 6:41pm

Question 1

0 / 1 pts

Consider the transition system in the following image, which option contains all the fluent(s)?



ɔu Answered

p, a, t, and f

Recall that a transition system is a directed graph whose vertices correspond to the states of the world and whose edges are labelled by actions. Thus a is an action. Besides, t and f are the values of fluent p and action a. t and f themselves are not fluent.

ɔrrect Answer

p

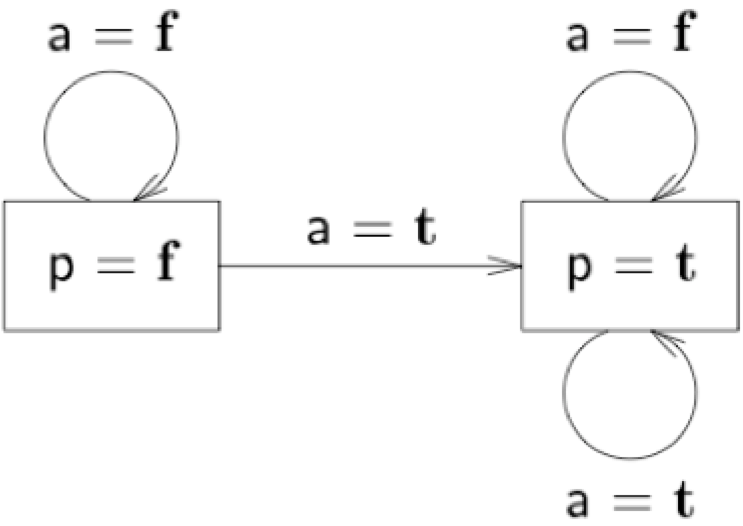
both p=t and a=t

a

Question 2

0.67 / 1 pts

Which options are the stable models that correspond to the following transition system? Choose all that apply.



Correct!

- [x] {p(f,0), p(f,1)}

This stable model means:

at time 0, p=f and a is NOT executed (i.e., a=f);

at time 1, p=f.

This follows the transition system in the given image.

- [ ] {p(f,0), p(t,1)}

Correct!

- [x] {p(f,0), p(t,1), a(0)}

This stable model means:

at time 0, p=f and a is executed;

at time 1, p=t.

This follows the transition system in the given image.

orrect Answer

- [ ] {p(t,0), p(t,1), a(0)}

Inanswered

Question 3

0 / 1 pts

Consider the same transition system in question 2, which option, along with the following constraint, represents the commonsense law of inertia?

:- T=1..m, not 1{p(t, T); p(f, T)}1.

- 1{p(t,0); p(f,0)}1.

orrect Answer

- {p(t,1)} :- p(t,0).
- p(t,1) :- a(0).
- {a(0)}.

Question 4

0 / 1 pts

Choose the transition system described by the following ASP program. Note that: p denotes a fluent symbol, and a denotes an action symbol. (Hint: the 4th rule is different from that in the slides.)

boolean(t;f).

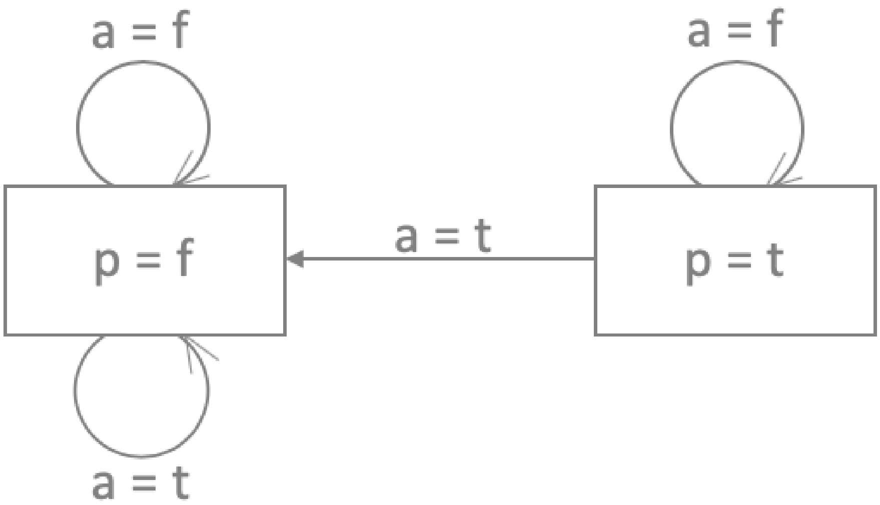
p(t,T+1) :- a(T), T=0..m-1.

1{p(B,0):boolean(B)}1.

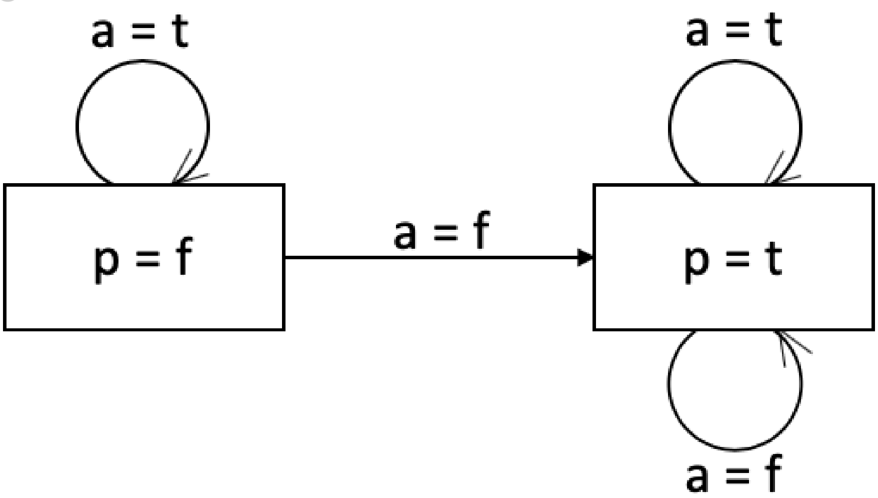
1{p(B,T):boolean(B)}1 :- T=1..m.

{a(T)} :- T=0..m-1.

{p(B,T+1)} :- p(B,T), T=0..m-1.



ou Answered



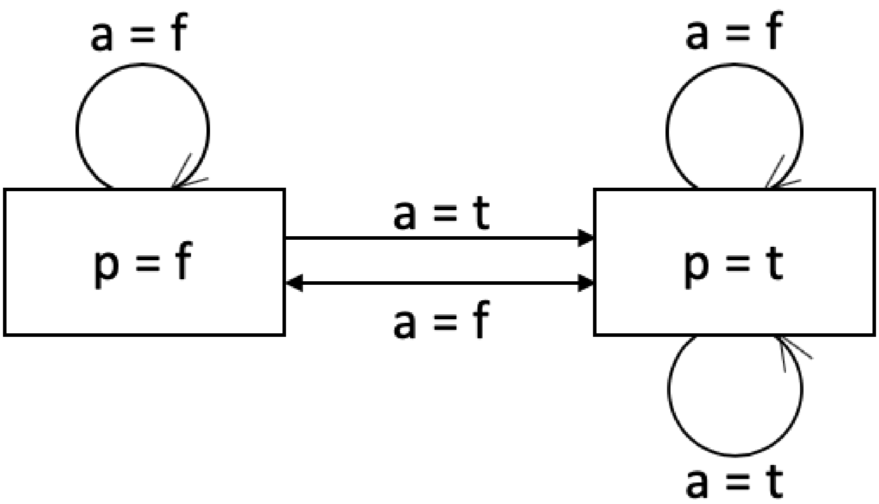
Let's first understand the meaning of each rule in the ASP program.

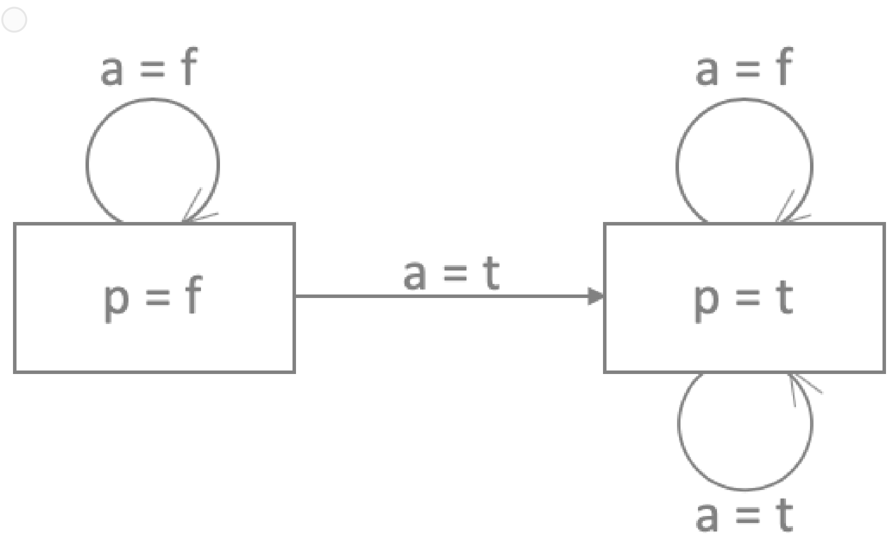
rule	**meaning**
boolean(t;f).	There are 2 Boolean values t and f.
p(t,T+1) :- a(T), T=0..m-1.	For any time stamps from 0 to m-1, if action a is executed, then the state at the next time stamp is p=t.
1{p(B,0):boolean(B)}1.	At time 0, we randomly choose a state from {p=t, p=f}.
1{p(B,T):boolean(B)}1 :- T=1..m.	At time 1 to m, we randomly choose a state from {p=t, p=f}.
{a(T)} :- T=0..m-1.	At time 0 to m-1, we randomly choose to execute action a or not.
{p(B,T+1)} :- p(B,T), T=0..m-1.	At time 0 to m-1, if p=B is true, p=B could be true at the next time stamp.

Apparently, there are 2 states: p=t and p=f.

Consider the case when p=t and a=f. The next state can be either p=t or p=f due to the 4th rule "1{p(B,T):boolean(B)}1 :- T=1..m."

orrect Answer





Question 5

0 / 1 pts

Consider the following clingo program for “Monkey and Bananas” problem.

```
%% sort and object declaration

boolean(t;f).
object(monkey;bananas;box).
location(l1;l2;l3).

%% state constraints

loc(bananas,L,T) :- hasBananas(t,T), loc(monkey,L,T).
loc(monkey,L,T) :- onBox(t,T), loc(box,L,T).

%% effect and preconditions of actions

% walk
loc(monkey,L,T+1) :- walk(L,T).
:- walk(L,T), loc(monkey,L,T).
:- walk(L,T), onBox(t,T).

% pushBox
loc(box,L,T+1) :- pushBox(L,T).
loc(monkey,L,T+1) :- pushBox(L,T).
:- pushBox(L,T), loc(monkey,L,T).
:- pushBox(L,T), onBox(t,T).
:- pushBox(L,T), loc(monkey,L1,T), loc(box,L2,T), L1 != L2.
```

```
% climbOn
onBox(t,T+1) :- climbOn(T).
:- climbOn(T), onBox(t,T).
:- climbOn(T), loc(monkey,L1,T), loc(box,L2,T), L1 != L2.

% climbOff
onBox(f,T+1) :- climbOff(T).
:- climbOff(T), onBox(f,T).

% graspBananas
hasBananas(t,T+1) :- graspBananas(T).
:- graspBananas(T), hasBananas(t,T).
:- graspBananas(T), onBox(f,T).
:- graspBananas(T), loc(monkey,L1,T), loc(bananas,L2,T), L1 != L2.

% disallow concurrent actions
:- walk(L,T), pushBox(L,T).
:- walk(L,T), climbOn(T).
:- pushBox(L,T), climbOn(T).
:- climbOff(T), graspBananas(T).

%% domain independent axioms

% fluents are initially exogenous
1{hasBananas(BB,0):boolean(BB)}1.
1{onBox(BB,0):boolean(BB)}1.
1{loc(O,LL,0):location(LL)}1 :- object(O).

% uniqueness and existence of fluent values
:- not 1{loc(O,LL,T)}1, object(O), T = 1..m.
:- not 1{onBox(BB,T)}1, T = 1..m.
:- not 1{hasBananas(BB,T)}1, T = 1..m.

% actions are exogenous
{walk(L,T)} :- location(L), T = 0..m-1.
{pushBox(L,T)} :- location(L), T = 0..m-1.
{climbOn(T)} :- T = 0..m-1.
{climbOff(T)} :- T = 0..m-1.
{graspBananas(T)} :- T = 0..m-1.

% commonsense law of inertia
{hasBananas(B,T+1)} :- hasBananas(B,T), T=0..m-1.
{onBox(B,T+1)} :- onBox(B,T), T=0..m-1.
{loc(O,L,T+1)} :- loc(O,L,T), T=0..m-1.
```

Which option is a set of rules that makes the following rule redundant?

:- climbOn(T), climbOff(T).

onBox(t,T+1) :- climbOn(T).

:- climbOn(T), onBox(t,T).

onBox(f,T+1) :- climbOff(T).

:- climbOff(T), onBox(f,T).

orrect Answer

onBox(t,T+1) :- climbOn(T).

onBox(f,T+1) :- climbOff(T).

:- not 1{onBox(BB,T)}1, T = 1..m.

ou Answered

onBox(t,T+1) :- climbOn(T).

:- climbOn(T), onBox(t,T).

:- climbOn(T), loc(monkey,L1,T), loc(box,L2,T), L1 != L2.

onBox(f,T+1) :- climbOff(T).

:- climbOff(T), onBox(f,T).

To make the rule ":- climbOn(T), climbOff(T)." redundant, we need to make sure that the case "both climbOn(T) and climbOff(T) are true" will never happen given the rules in this option. In other words, we need to show that, with the rules in this option, if both climbOn(T) and climbOff(T) are true, either we can derive

$$\perp$$

or some cardinality constraint is not satisfied. However, the only thing we can derive is "onBox(t, T+1) and onBox(f, T+1) are true."

Given

climbOn(T), climbOff(T), onBox(t, T+1), onBox(f, T+1)

we cannot find a constraint whose body is satisfied, thus we cannot derive

$$\perp$$

from the constraints in this option.

onBox(t,T+1) :- climbOn(T).

onBox(f,T+1) :- climbOff(T).

1{onBox(BB,0):boolean(BB)}1.

Question 6

0 / 1 pts

Consider the following clingo program for “Blocks World” problem.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% sort and object declaration
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% every block is a location
location(B) :- block(B).

% the table is a location
location(table).

%%%%%%%%%%%%%%%%%%%%%%%%%%
% state description
%%%%%%%%%%%%%%%%%%%%%%%%%%

% two blocks can't be on the same block at the same time
:- 2{on(BB,B,T)}, block(B), T = 0..m.


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% effect and preconditions of action
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% effect of moving a block
on(B,L,T+1) :- move(B,L,T).

% concurrent actions are limited by num of grippers
:- not {move(BB,LL,T)} grippers, T = 0..m-1.

% a block can be moved only when it is clear
:- move(B,L,T), on(B1,B,T).

% a block can't be moved onto a block that is being moved also
:- move(B,B1,T), move(B1,L,T).
```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% domain independent axioms
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% fluents are initially exogenous
1{on(B,LL,0):location(LL)}1 :- block(B).

% uniqueness and existence of value constraints
:- not 1{on(B,LL,T)}1, block(B), T=1..m.

% actions are exogenous
{move(B,L,T)} :- block(B), location(L), T = 0..m-1.

% commonsense law of inertia
{on(B,L,T+1)} :- on(B,L,T), T < m.

Suppose we have two grippers G1 and G2. According to the ASP formalization, can G1 move B onto C at the same time G2 moves C onto the table?

orrect Answer

No

ou Answered

Yes

Consider the following rule when B=b, B1=c, L=table, and T=t.

% a block can't be moved onto a block that is being moved also

:- move(B,B1,T), move(B1,L,T).

Question 7

0 / 1 pts

Consider the following clingo program for “Blocks World” problem.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% sort and object declaration
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% every block is a location
location(B) :- block(B).

% the table is a location
location(table).

%%%%%%%%%%%%%%%%%%%%%%%%%%
% state description
%%%%%%%%%%%%%%%%%%%%%%%%%%

% two blocks can't be on the same block at the same time
:- 2{on(BB,B,T)}, block(B), T = 0..m.


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% effect and preconditions of action
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% effect of moving a block
on(B,L,T+1) :- move(B,L,T).

% concurrent actions are limited by num of grippers
:- not {move(BB,LL,T)} grippers, T = 0..m-1.

% a block can be moved only when it is clear
:- move(B,L,T), on(B1,B,T).

% a block can't be moved onto a block that is being moved also
:- move(B,B1,T), move(B1,L,T).


%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% domain independent axioms
%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% fluents are initially exogenous
1{on(B,LL,0):location(LL)}1 :- block(B).

% uniqueness and existence of value constraints
:- not 1{on(B,LL,T)}1, block(B), T=1..m.

% actions are exogenous
{move(B,L,T)} :- block(B), location(L), T = 0..m-1.

% commonsense law of inertia
{on(B,L,T+1)} :- on(B,L,T), T < m.
```

Suppose we have two grippers G1 and G2. According to the ASP formalization, can G1 move B onto D at the same time G2 moves C onto the table?

ou Answered

No

The question is asking the possibility of both move(b, d , t) and move(c, table, t) are true for some t in {0, ..., m-1}. You can read through all the constraints and see if any constraint can be violated by these 2 atoms. If you cannot find such a constraint, then these 2 atoms are actually possible to be true for some t.

orrect Answer

Yes

Question 8

1 / 1 pts

Consider the following clingo program for “Blocks World” problem.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% sort and object declaration
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% every block is a location
location(B) :- block(B).

% the table is a location
location(table).

%%%%%%%%%%%%%%%%%%%%%%%%%%
% state description
%%%%%%%%%%%%%%%%%%%%%%%%%%

% two blocks can't be on the same block at the same time
:- 2{on(BB,B,T)}, block(B), T = 0..m.


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% effect and preconditions of action
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

% effect of moving a block
on(B,L,T+1) :- move(B,L,T).

% concurrent actions are limited by num of grippers
:- not {move(BB,LL,T)} grippers, T = 0..m-1.

% a block can be moved only when it is clear
:- move(B,L,T), on(B1,B,T).

% a block can't be moved onto a block that is being moved also
:- move(B,B1,T), move(B1,L,T).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% domain independent axioms
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% fluents are initially exogenous
1{on(B,LL,0):location(LL)}1 :- block(B).

% uniqueness and existence of value constraints
:- not 1{on(B,LL,T)}1, block(B), T=1..m.

% actions are exogenous
{move(B,L,T)} :- block(B), location(L), T = 0..m-1.

% commonsense law of inertia
{on(B,L,T+1)} :- on(B,L,T), T < m.

Suppose there are 2 boxes A and B where A is on the table and B is on A. If you want to use clingo to generate a plan to move A onto B, which option is the set of rules you need to add?

- on(a, table, 0; b, a, 0; a, b, m; b, table, m).

Correct!

- :- not on(a, table, 0).

  :- not on(b, a, 0).

  :- not on(a, b, m).

  :- not on(b, table, m).

In clingo, if we want to represent the initial state, we can use either facts in the form of "ATOM." or constraints in the form of ":- not ATOM."

If we want to represent the goal state, we MUST use constraints in the form of ":- not ATOM." where the ATOM must represent something in the last time stamp m.

:- not on(a, table, 0; b, a, 0).

- on(a, b, m; b, table, m).
- :- not on(a, table, 0; b, a, 0; a, b, 1; b, table, 1).

Question 9

0 / 1 pts

Consider the following clingo program discussed in the module “Expressive Possibilities”.

```
boolean(t;f).

% sorts and object declarations
object(jack;car).
location(home;work).

% effect and precondition of go
loc(jack,L,T+1) :- go(L,T).
{loc(car,L,T+1)} :- go(L,T), loc(car,L1,T), loc(jack,L1,T), T=0..m-1.
:- go(L,T), loc(jack,L,T).

% fluents are exogenous initially
1{loc(O,LL,0):location(LL)}1 :- object(O).

% uniqueness and existence of fluent values
:- not 1{loc(O,LL,T)}1, object(O), T=1..m.

% actions are exogenous
{go(L,T)} :- location(L), T=0..m-1.

% fluents are inertial
{loc(O,L,T+1)} :- loc(O,L,T), T=0..m-1.
```

Which option correctly explains why the effect of action “go” is nondeterministic?

The rule below defines a precondition of action “go” that jack cannot go to the place that he is currently at.

:- go(L,T), loc(jack,L,T).

ou Answered

The rule below represents that the action “go” is exogenous, which means it is not determined.

{go(L,T)} :- location(L), T=0..m-1.

This rule does not talk about the effect of action "go".

orrect Answer

The rule below uses a choice rule in the head to represent that the location of the car may or may not change to the location that jack goes to.

{loc(car,L,T+1)} :- go(L,T), loc(car,L1,T), loc(jack,L1,T), T=0..m-1.

The rule below is nondeterministic since the location of O may or may not be L even if it was at L.

{loc(O,L,T+1)} :- loc(O,L,T), T=0..m-1.

Question 10

0 / 1 pts

Consider the following clingo program for “pendulum” problem.

% sorts and object declaration
boolean(t;f).

% effect of hold
right(T+1) :- hold(T), right(T).

left(T+1) :- hold(T), left(T).

% by default, pendulum changes the position
{left(T+1)} :- right(T), T=0..m-1.
{right(T+1)} :- left(T), T=0..m-1.

% fluents are exogenous initially
1{right(0);left(0)}1.

% uniqueness and existence of values for fluents
:- not 1{right(T);left(T)}1, T=1..m.

% exogenous action
{hold(T)} :- T=0..m-1.

Suppose we observe that the pendulum is on the right at time 0. We also observe that it is also on the right at time 2. When m equals to 2, how many stable models of the above program will agree with these observations?

- 0

ou Answered

- 4

Since the stable models of this clingo program represent the possible explanations to the problem, we count the number of explanations to the problem when there are only 2 actions at most (i.e., at most one happens at time stamp 0 and one happens at time stamp 1).

Since the initial state and the final state are the same, there are only 2 possible explanations: (i) there is no action, and (ii) we hold at both time stamp 0 and 1. Try to find if there is any other possible explanations and you will see that you cannot find more.

- 1

orrect Answer

- 2

Module 6 Graded Quiz

Due Apr 10 at 11:59pm **Points** 10 **Questions** 10
Available Mar 27 at 12am - Apr 10 at 11:59pm 15 days
Time Limit 300 Minutes **Allowed Attempts** 3

This quiz was locked Apr 10 at 11:59pm.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	19 minutes	9 out of 10

Score for this attempt: **9** out of 10
Submitted Apr 10 at 9:56am
This attempt took 19 minutes.

Question 1

1 / 1 pts

Consider the following **MLN** program with two object constants alice and bob. Compute the unnormalized probability measure (i.e., weight) of the interpretation I = {smoke(alice), influence(alice, bob), smoke(bob)} under **MLN**. (x, y range over both object constants.)

2: smoke(y) $\leftarrow$ smoke(x) $\wedge$ influence(x,y)

1: smoke(alice)

α: influence(alice, bob)

- exp(1)
- exp(3)
- exp(α+3)
- Correct! exp(α+9)

Question 2

1 / 1 pts

Consider the following **LPMLN** program with two object constants alice and bob. Compute the unnormalized probability measure (i.e., weight) of the interpretation I = {smoke(alice), influence(alice, bob), smoke(bob)} under **LPMLN**. (x, y range over both object constants.)

2: smoke(y) $\leftarrow$ smoke(x) $\wedge$ influence(x,y)

1: smoke(alice)

α: influence(alice, bob)

- 0
- Correct! exp(α+9)
- exp(α+3)
- exp(α+1)

Question 3

1 / 1 pts

Consider the following LPMLN program. Apply the completion method to convert the LPMLN program into an equivalent Markov Logic Network. Which option is the correct MLN program?

1: p $\leftarrow \neg$q

2: q $\leftarrow \neg$p

4: q $\leftarrow \neg$r

1: $p \leftarrow \neg q$

2: $q \leftarrow \neg p$

4: $q \leftarrow \neg r$

α: $p \rightarrow \neg q$

α: $q \rightarrow \neg p$

α: $q \rightarrow \neg r$

Correct!

1: $p \leftarrow \neg q$

2: $q \leftarrow \neg p$

4: $q \leftarrow \neg r$

α: $p \rightarrow \neg q$

α: $q \rightarrow \neg p \vee \neg r$

α: $r \rightarrow \bot$

1: $p \leftarrow \neg q$

2: $q \leftarrow \neg p$

4: $q \leftarrow \neg r$

α: $p \rightarrow \neg q$

α: $q \rightarrow \neg p \vee \neg r$

1: $p \leftarrow \neg q$

2: $q \leftarrow \neg p$

4: $q \leftarrow \neg r$

α: $p \rightarrow \neg q$

α: $q \rightarrow \neg p$

α: $q \rightarrow \neg r$

α: $r \rightarrow \bot$

Question 4

1 / 1 pts

Which option is the most probable stable model of the following LPMLN program?

1: p ← ¬q

α: q ← ¬p

-4: p← ¬r

- ∅

Correct!

- {q}
- {p, q}
- {p}

Question 5

1 / 1 pts

Which option is the weight of the most probable stable model of the following LPMLN program?

2: p ← q

-2: q ← ¬r

-5: ⊥ ← p

Correct!

- exp(0)
- exp(2)
- 2
- -5

Question 6

0 / 1 pts

Consider the following LPMLN program. Translate it into an ASP program containing weak constraints. Which option is the correct translated ASP program?

10: q ← p

1: r ← p

5: p

-20: ⊥ ← ¬r

10: q ← p

1: r ← p

5: p

-20: ⊥ ← ¬r

α: p → ⊤

α: q → p

α: r → p

ou Answered

unsat(1) :- p, not q.

q :- p, not unsat(1).

:~ unsat(1). [-10]

unsat(2) :- p, not r.

r :- p, not unsat(2).

:~ unsat(2). [-1]

unsat(3) :- not p.

p :- not unsat(3).

:~ unsat(3). [-5]

unsat(4) :- not r.

:- not r, not unsat(4).

:~ unsat(4). [20]

orrect Answer

unsat(1) :- p, not q.

q :- p, not unsat(1).

:~ unsat(1). [10]

unsat(2) :- p, not r.

r :- p, not unsat(2).

:~ unsat(2). [1]

unsat(3) :- not p.

p :- not unsat(3).

:~ unsat(3). [5]

unsat(4) :- not r.

:- not r, not unsat(4).

:~ unsat(4). [-20]

unsat(1) :- p, not q.

:~ unsat(1). [-10]

unsat(2) :- p, not r.

:~ unsat(2). [-1]

unsat(3) :- not p.

:~ unsat(3). [-5]

unsat(4) :- not r.

:~ unsat(4). [20]

Question 7

1 / 1 pts

Consider the following LPMLN program. Which option is the most probable stable model of the program?

10: q ← p

1: r ← p

5: p

-20: ⊥ ← ¬r

- {p, q, r}
- Correct! {p, q}
- {p}
- {q}

Question 8

1 / 1 pts

Which option about probability is impossible no matter what random variables A and B are?

- P(A | B) < P(A)
- P(A | B) > P(A)
- P(A, B) < P(A)
- Correct! P(A, B) > P(A)

Question 9

1 / 1 pts

Consider the following full joint distribution for Boolean variables A, B, and C. Which option is closest to the value of P(A = 0 | B = 1)?

A	B	C	P(A,B,C)
0	0	0	0.03
0	0	1	0.12
0	1	0	0.17
0	1	1	0.18
1	0	0	0.03
1	0	1	0.12
1	1	0	0.24
1	1	1	0.11

- 0

Correct!

- 0.5
- 0.71
- 0.35

Question 10

1 / 1 pts

Consider the same full joint distribution for Boolean variables A, B, and C as in question 9. Are A and B independent of each other?

A	B	C	P(A,B,C)
0	0	0	0.03
0	0	1	0.12
0	1	0	0.17
0	1	1	0.18
1	0	0	0.03
1	0	1	0.12
1	1	0	0.24
1	1	1	0.11

Correct!

- True
- False

Quiz Score: **9** out of 10

Module 6 Practice Quiz

Due No due date **Points** 10 **Questions** 10
Available after Mar 20 at 12am **Time Limit** None
Allowed Attempts Unlimited

Take the Quiz Again

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	less than 1 minute	4 out of 10

Submitted May 3 at 6:58pm

Question 1

0 / 1 pts

Consider the following **MLN** program with two object constants alice and bob. Compute the unnormalized probability measure (i.e., weight) of the interpretation I = {smoke(alice), smoke(bob)} under **MLN**. (x, y range over both object constants.)

$$2: \; smoke(y) \leftarrow smoke(x) \wedge influence(x, y)$$

$$1: \; smoke(alice)$$

$$\alpha: \; influence(alice, bob)$$

0

ou Answered

exp(1)

For any MLN program Π, every interpretation I of Π is a model of Π. And the weight of I is e to the power of S, where S is the sum of the weights of the **grounded rules** that are **satisfied** by I.

First, we need to ground the given MLN program into the following grounded counter-part.

$2: \; smoke(alice) \leftarrow smoke(alice) \wedge influence(alice, alice)$

$2: \; smoke(bob) \leftarrow smoke(alice) \wedge influence(alice, bob)$

$2: \; smoke(alice) \leftarrow smoke(bob) \wedge influence(bob, alice)$

$2: \; smoke(bob) \leftarrow smoke(bob) \wedge influence(bob, bob)$

$1: \; smoke(alice)$

$\alpha: \; influence(alice, bob)$

Second, we need to figure it out which of the above grounded rules are satisfied by$I = \{smoke(alice), smoke(bob)\}$

Third, we need to accumulate the weights of the satisfied grounded rules. If the accumulated value is S, then the weight of I is then exp(S).

orrect Answer

- exp(9)
- exp(3)

Question 2

0 / 1 pts

Consider the following **LPMLN** program with two object constants alice and bob. Compute the unnormalized probability measure (i.e., weight) of the interpretation I = {smoke(alice), smoke(bob)} under **LPMLN**. (x, y range over both object constants.)

$2: \; smoke(y) \leftarrow smoke(x) \wedge influence(x, y)$

$1: \; smoke(alice)$

$\alpha: \; influence(alice, bob)$

- exp(α+1)

ou Answered

exp(3)

For any **LPMLN** program Π and any interpretation I of Π it's possible that I is not a stable model of Π, in which case, the weight of I is 0. This is different from **MLN** since the weight of any interpretation I under **MLN** cannot be 0.

Below are the steps to compute the weight of an interpretation I ofΠ under **LPMLN**. It is very similar to that under **MLN** except for a stable model checking process in step 3.

First, we need to ground the given **LPMLN** program into the following grounded counter-part.

$2:\ smoke(alice) \leftarrow smoke(alice) \wedge influence(alice, alice)$

$2:\ smoke(bob) \leftarrow smoke(alice) \wedge influence(alice, bob)$

$2:\ smoke(alice) \leftarrow smoke(bob) \wedge influence(bob, alice)$

$2:\ smoke(bob) \leftarrow smoke(bob) \wedge influence(bob, bob)$

$1:\ smoke(alice)$

$\alpha:\ influence(alice, bob)$

Second, we need to figure it out which of the above grounded rules are satisfied by $I = \{smoke(alice), smoke(bob)\}$. The satisfied rules are as follows.

$2:\ smoke(alice) \leftarrow smoke(alice) \wedge influence(alice, alice)$

$2:\ smoke(bob) \leftarrow smoke(alice) \wedge influence(alice, bob)$

$2:\ smoke(alice) \leftarrow smoke(bob) \wedge influence(bob, alice)$

$2:\ smoke(bob) \leftarrow smoke(bob) \wedge influence(bob, bob)$

$1:\ smoke(alice)$

Third, we check if I is a stable model of the above set of satisfied grounded rules (without the weights). If I is not a stable model, then the weight of I is 0. If I is a stable model, then we simply accumulate the weights of the above rules, and if the accumulated value is S, the weight of I is exp(S).

Hint: you need to check whether I is a stable model of the following ASP program Π', which is the unweighted part of the satisfied grounded rules.

$smoke(alice) \leftarrow smoke(alice) \wedge influence(alice, alice)$

$smoke(bob) \leftarrow smoke(alice) \wedge influence(alice, bob)$

$smoke(alice) \leftarrow smoke(bob) \wedge influence(bob, alice)$

$smoke(bob) \leftarrow smoke(bob) \wedge influence(bob, bob)$

$smoke(alice)$

You can check whether I is a stable model of Π' according to the definition of a stable model of an ASP program by constructing the reduct of Π' w.r.t. I.

Alternatively, an easier way is to find the stable model(s) of Π' by intuition. We know $smoke(alice)$ must be true, but there is no support for us to believe any other atom. Thus the stable model of $\Pi'$is $\{smoke(alice)\}$.

orrect Answer

- 0
- exp(α+9)

Question 3

0 / 1 pts

Consider the following LPMLN program. Apply the completion method to convert the LPMLN program into an equivalent Markov Logic Network. Which option is the correct MLN program?

$\alpha : p$

$1 : p \leftarrow \neg q$

ou Answered

- $\alpha : p$
  $1 : p \leftarrow \neg q$
  $\alpha : p \rightarrow \neg q$
  $\alpha : q \rightarrow \perp$

Given any LPMLN program Π, we can apply the completion method to turn Π into its MLN counter-part Π'. The steps are as follows.

First, Π' contains all rules in Π.

Second, for every **grounded** atom A in Π, Π' contains the following rule

$$\alpha : \ A \rightarrow \bigvee_{w:A \leftarrow Body \in \Pi} Body.$$

In this example, rule p can be treated as

$p \leftarrow \top$.

Thus for grounded atom p, there should be a rule in the form of

$$\alpha : \ p \rightarrow \bigvee_{w:p \leftarrow Body \in \Pi} Body$$

where the right-hand side of the symbol $\rightarrow$ is the disjunction of the body of 2 rules. However, the option does not consider the body of rule $p \leftarrow \top$.

$\alpha : p$
$1 : p \leftarrow \neg q$
$\alpha : p \rightarrow \top \vee \neg q$

orrect Answer

$\alpha : p$
$1 : p \leftarrow \neg q$
$\alpha : p \rightarrow \top \vee \neg q$
$\alpha : q \rightarrow \bot$

$\alpha : p$
$1 : p \leftarrow \neg q$
$\alpha : p \rightarrow \top$
$\alpha : p \rightarrow \neg q$
$\alpha : q \rightarrow \bot$

Question 4

1 / 1 pts

Which option is the most probable stable model of the following LPMLN program?

$1 : p$

$\alpha : p \leftarrow q$

$\alpha : \bot \leftarrow \neg q$

- {p, q}
- {q}

Correct!

- {p}

The most probable stable model is the stable model with the highest probability, which is also the stable model with the highest weight. Thus, normally, we need to compute the weight of all 4 interpretations of this program: $\emptyset, \{p\}, \{q\}, \{p, q\}$

One trick here is that, from intuition, we know $q$can never be in any stable model since there is no support for$q$ (i.e., there is no rule whose head is q). Thus we only need to compute the weights of $\emptyset$ and $\{p\}$.

The only rule that is satisfied by $\emptyset$ is

$\alpha : p \leftarrow q$

Since $\emptyset$is the stable model of $p \leftarrow q$, the weight of $\emptyset$ is e^{α}

The rules that are satisfied by$\{p\}$ are

$1 : p$

$\alpha : p \leftarrow q$

Since $\{p\}$is the stable model of

p

$p \leftarrow q$

the weight of $\{p\}$is $e^{1+\alpha}$.

- $\emptyset$

Question 5

0 / 1 pts

Which option is the weight of the most probable stable model of the following LPMLN program?

$$1 : q$$
$$-2 : p \leftarrow q$$
$$-3 : \bot \leftarrow r$$

ou Answered

exp(1)

The most probable stable model is the stable model with the highest probability, which is also the stable model with the highest weight. Thus, normally, we need to compute the weight of all $2^3 = 8$ interpretations of this program.

One trick here is that, from intuition, we know r can never be in any stable model since there is no support for r (i.e., there is no rule whose head is r). Thus we only need to compute the weights of $\varnothing, \{p\}, \{q\}, \{p, q\}$

From this point, you may either compute the weights of these 4 interpretations and find the highest one, or try to guess the stable model with maximal possible weight.

How to guess? Well, since r must be false, the 3rd rule must be satisfied. Thus the best possible weight is e to the power of $1 - 3 = -2$ in the case when the 1st rule is satisfied and the 2nd rule is not satisfied. However, this option is already better than the highest possible weight.

orrect Answer

exp(-2)

-2

1

Question 6

1 / 1 pts

Consider the following LPMLN program. Translate it into an ASP program containing weak constraints. Which option is the correct translated ASP program?

$1 : p$

$-2 : p \leftarrow q$

$3 : \bot \leftarrow p$

Correct!

unsat(1) :- not p.

p :- not unsat(1).

:~ unsat(1). [1]

unsat(2) :- q, not p.

p :- q, not unsat(2).

:~ unsat(2). [-2]

unsat(3) :- p.

:- p, not unsat(3).

:~ unsat(3). [3]

If an LPMLN program Π only contains soft rules, then its translation to an ASP program Π' with weak constraints is defined as follows.

For any rule $w : head \leftarrow body$ of index$i$ in Π, Π' contains the following 3 rules under the syntax of clingo.

unsat(i) :- body, not head.

head :- body, not unsat(i).

:~ unsat(i). [w]

For example, in this question, the first rule can be seen as

$1 : p \leftarrow \top$

and its translated ASP counter-part is

unsat(1) :- not p.

p :- not unsat(1).

:~ unsat(1). [1]

$1 : \mathrm{p}$
$-2 : \mathrm{p} \leftarrow \mathrm{q}$
$3 : \perp \leftarrow \mathrm{p}$
$\alpha : \mathrm{p} \rightarrow \top \vee \mathrm{q}$
$\alpha : q \rightarrow \perp$

unsat(1) :- not p.

p :- not unsat(1).

:~ unsat(1). [-1]

unsat(2) :- q, not p.

p :- q, not unsat(2).

:~ unsat(2). [2]

unsat(3) :- p.

:- p, not unsat(3).

:~ unsat(3). [-3]

unsat(1) :- not p.

:~ unsat(1). [1]

unsat(2) :- q, not p.

:~ unsat(2). [-2]

unsat(3) :- p.

:~ unsat(3). [3]

Question 7

0 / 1 pts

Consider the following LPMLN program. Which option is the most probable stable model of the program?

$1 : p$

$-2 : p \leftarrow q$

$3 : \perp \leftarrow p$

ou Answered

{p}

The most probable stable model is the stable model with the highest probability, which is also the stable model with the highest weight. Thus, normally, we need to compute the weight of all 4 interpretations of this program: $\varnothing, \{p\}, \{q\}, \{p, q\}$

One trick here is that, from intuition, we know q can never be in any stable model since there is no support for q (i.e., there is no rule whose head is q). Thus we only need to compute the weights of $\varnothing$ and $\{p\}$

The rules that are satisfied by $\varnothing$ are

$-2 : p \leftarrow q$

$3 : \perp \leftarrow p$

and the rules that are satisfied by $\{p\}$ are

$1 : p$

$-2 : p \leftarrow q$

Try to figure it out what are the weights of $\varnothing$ and $\{p\}$. Don't forget to check whether they are the stable models of the satisfied rules (without weights).

{p, q}

{q}

orrect Answer

$\varnothing$

Question 8

0 / 1 pts

Which option about probability is impossible no matter what random variables A and B are?

P(A | B) < P(B | A)

ou Answered

P(A, B) = P(A | B)

P(A|B) = P(A,B) / P(B)

If P(B)=1, then the equation in this option is true.

orrect Answer

P(A | B) < P(A, B)

P(A, B) < P(B)

Question 9

1 / 1 pts

Consider the following full joint distribution for Boolean variables A, B, and C. Which option is closest to the value of P(A = 1 | B=1, C = 0)?

A	B	C	P(A,B,C)
0	0	0	0.2
0	0	1	0.1
0	1	0	0.15
0	1	1	0.05
1	0	0	0.05
1	0	1	0.25
1	1	0	0.05
1	1	1	0.15

0.05

2.5

Correct!

0.25

P(A = 1 | B=1, C = 0) = P(A = 1, B=1, C = 0) / P(B=1, C = 0)

P(A = 1, B=1, C = 0) = 0.05

P(B=1, C = 0) = P(A = 0, B=1, C = 0) + P(A = 1, B=1, C = 0) = 0.15 + 0.05 = 0.2

Thus P(A = 1 | B=1, C = 0) = 0.05/0.2 = 0.25

0.5

Question 10

1 / 1 pts

Consider the same full joint distribution for Boolean variables A, B, and C as in question 9. Are A and B independent of each other?

A	B	C	P(A,B,C)
0	0	0	0.2
0	0	1	0.1
0	1	0	0.15
0	1	1	0.05
1	0	0	0.05
1	0	1	0.25
1	1	0	0.05
1	1	1	0.15

Correct!

True

In this question, A is independent of B iff

P(A=0) = P(A=0 | B=0) = P(A=0 | B=1), and

P(A=1) = P(A=1 | B=0) = P(A=1 | B=1)

Furthermore, we compute the value of each of them.

P(A=0) = 0.2 + 0.1+ 0.15 + 0.05 = 0.5

P(A=1) = 1 - P(A=0) = 0.5

P(A=0 | B=0) = P(A=0, B=0) / P(B=0) = 0.3/0.6 = 0.5

P(A=1 | B=0) = 1 - P(A=0 | B=0) = 0.5

P(A=0 | B=1) = P(A=0, B=1) / P(B=1) = 0.2/0.4 = 0.5

P(A=1 | B=1) = 1 - P(A=0 | B=1) = 0.5

False

Module 7 Graded Quiz

Due Apr 24 at 11:59pm **Points** 10 **Questions** 10
Available Mar 31 at 12:59am - Apr 24 at 11:59pm 25 days
Time Limit 300 Minutes **Allowed Attempts** 3

This quiz was locked Apr 24 at 11:59pm.

Attempt History

	Attempt	Time	Score
KEPT	Attempt 1	3 minutes	10 out of 10
LATEST	Attempt 2	300 minutes	0 out of 10
	Attempt 1	3 minutes	10 out of 10

Score for this attempt: **0** out of 10
Submitted Apr 21 at 6:44pm
This attempt took 300 minutes.

Inanswered

Question 1

0 / 1 pts

Consider the following TBox.

$\forall R.\, A \sqsubseteq \exists R.\, C$

Which option is the First-Order formula that is translated from this TBox?

- $\forall y(R(x,y) \wedge A(y)) \to \exists y(R(x,y) \wedge C(y))$
- $\forall y(R(x,y) \wedge A(y)) \to \exists y(R(x,y) \wedge C(y))$
- orrect Answer: $\forall x(\forall y(R(x,y) \to A(y)) \to \exists y(R(x,y) \wedge C(y)))$
- $\forall y(R(x,y) \to A(y)) \to \exists y(R(x,y) \wedge C(y))$

Inanswered

Question 2

0 / 1 pts

Which option **entails** the following TBox?

$\forall R.\, A \sqsubseteq \exists R.\, B$

- $\exists R.\, A$
- $\forall R.\, A$
- $A \sqsubseteq B$
- orrect Answer $\exists R.\, B$

Inanswered

Question 3

0 / 1 pts

Consider the following ALC concepts.

hasChild(Joe, Ahn)

hasChild(Joe, Eva)

hasChild(Joe, Mary)

≤2 hasChild(Joe)

Which option is correct?

- It is an ABox and is not satisfiable.
- It is a TBox and is satisfiable.
- orrect Answer It is an ABox and is satisfiable.
- It is a TBox and is not satisfiable.

Inanswered

Question 4

0 / 1 pts

Which kind of reasoning in ontologies should you use if you are asked “whether Barack’s last name is Obama” with some knowledge about Barack?

- Concept satisfiability
- Answering concept queries
- Taxonomies

orrect Answer

- Instance checking

Inanswered

Question 5

0 / 1 pts

Which option is a suitable ALC concept in description logic for the class of humans “pet owner who only owns cat”, using the following concept names and role name?

concept names: Person, Cat, Animal

role name: owns

orrect Answer

- Person ⊓ ∃owns.Animal ⊓ ∀owns .Cat
- Person ⊓ Cat ⊓ owns(Person, Cat)
- Person ⊓ owns(Person, Cat)
- Person ⊓ ∀owns .Cat

Inanswered

Question 6

0 / 1 pts

Which option about the use of ontologies in industrial applications is correct?

- CYC is different from DBpedia in the sense that it emphasizes on individuals, not general knowledge about various domains.
- YAGO2 is built by a team of trained ontology engineers who organize the ontology and write the axioms.

orrect Answer

- The relationship between entities is represented by typed edges in a knowledge graph.
- WordNet is a knowledge graph and also an ontology constructed by human experts.

Inanswered

Question 7

0 / 1 pts

Which option is correct about RDFS?

orrect Answer

- Instances of a class inherit the properties of that class.
- “Only humans can teach a class” can be guaranteed by range restriction.
- “is taught by” is a subclass of “involves”
- The class hierarchy is a binary tree.

Unanswered

Question 8

0 / 1 pts

Using the following concept names and role names, which option can correctly capture the knowledge “drivers of cars are adults”?

concept names: Person, Car, Adult

role names: controls

- $\forall$controls.Car $\sqsubseteq$ Adult
  Adult $\sqsubseteq$ Person

Correct Answer

- Person $\sqcap$ $\exists$controls.Car $\sqsubseteq$ Adult
- Person $\sqcap$ controls.Car $\sqcap$ Adult
- Adult $\equiv$ Person $\sqcap$ $\forall$controls.Car

Unanswered

Question 9

0 / 1 pts

Using the following concept names and role names, which option can correctly capture the knowledge “cars have between three and four wheels”?

concept names: Car, Wheel

role names: hasPart

Correct Answer

- Car $\sqsubseteq$ ($\geq$3 hasPart.Wheel) $\sqcap$ ($\leq$4 hasPart.Wheel)
- Car $\equiv$ ($\geq$3 hasPart.Wheel)
  Car $\equiv$ ($\leq$4 hasPart.Wheel)
- Car $\sqcap$ $\forall$hasPart.Wheel $\sqsubseteq$ Wheel $\geq$ 3 $\sqcap$ Wheel $\leq$ 4
- Car $\sqsubseteq$ (3 $\leq$ hasPart.Wheel $\leq$ 4)

Unanswered

Question 10

0 / 1 pts

Which of the following statements is correct?

- A ⊓ ∃r.B is subsumed by A ⊓ ∃r.⊥
- B is subsumed by A⊔¬B
- A ⊓ ∃r.(B⊔C) is subsumed by A ⊓ ∃r.B
- Correct Answer: A⊓¬A is subsumed by B

Quiz Score: **0** out of 10

Module 7 Practice Quiz

Due No due date **Points** 10 **Questions** 10
Available after Apr 1 at 11:59pm **Time Limit** None
Allowed Attempts Unlimited

Take the Quiz Again

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	293 minutes	2 out of 10

Submitted May 3 at 7:07pm

Question 1

0 / 1 pts

Consider the following TBox.

$$A \sqsubseteq \exists R.B$$
$$A \sqsubseteq \forall R.C$$

Which option is the First-Order formula that is translated from this TBox?

ou Answered

$\exists x(A(x) \rightarrow \exists y(R(x,y) \land B(y))) \land \forall x(A(x) \rightarrow \forall y(R(x,y) \rightarrow C(y)))$

You may refer to Module "DL to FOL" for the formal translation. Alternatively, you can translate according to a proper way of "reading" this Tbox.

The first concept inclusion reads "for any x, if x belongs to the set A, then x must belong to the set of $\exists R.B$. The second concept inclusion reads "for any x, if x belongs to the set A, then x must belong to the set of $\forall R.C$". Thus you may notice that, a concept inclusion in Tbox is defining the relationship between 2 sets: for all x, if x belongs to the left-hand side of $\sqsubseteq$, then x must belong to the right-hand side of $\sqsubseteq$. So the outmost quantifier must be

$$\forall$$

instead of $\exists$.

Hint: "x belongs to the set of $\exists R.B$" means that there exists a y such that $R(x,y) \wedge B(y)$ is true. "x belongs to the set of $\forall R.C$" means that, for all y, if $R(x,y)$ is true, then $C(y)$is true.

orrect Answer

$\forall x(A(x) \rightarrow \exists y(R(x,y) \wedge B(y))) \wedge \forall x(A(x) \rightarrow \forall y(R(x,y) \rightarrow C(y)))$

$\forall x(A(x) \rightarrow \exists y(R(x,y) \wedge B(y))) \wedge \forall x(A(x) \rightarrow \forall y(R(x,y) \wedge C(y)))$

$\exists x(A(x) \rightarrow \exists y(R(x,y) \rightarrow B(y))) \wedge \forall x(A(x) \rightarrow \forall y(R(x,y) \rightarrow C(y)))$

Question 2

0 / 1 pts

Which option is entailed by the following TBox?

$$A \sqsubseteq \exists R.B$$
$$A \sqsubseteq \forall R.C$$

ou Answered

$B \sqsubseteq C$

To check the entailment, we need to check if every model of B is a model of$C$.

An easier way is to think of each concept inclusion as a FOL formula. The given Tbox can be translated into the following set of FOL formulas (which can be seen as a single FOL formula that is the conjunction of all the formulas in the set).

$\forall x(A(x) \to \exists y(R(x,y) \land B(y)))$

$\forall x(A(x) \to \forall y(R(x,y) \to C(y)))$

Thus, for all x, if $A(x)$ is true, then

- there exists a y such that $R(x,y)$ and $B(y)$ are true
- for all z if $$R(x,z)$is true, then$C(z)$ is true. Note that this$z$ could be the y in the previous bullet.

The option says: for all x, if $B(x)$ is true then$C(x)$ must be true. However, this cannot be derived since there is no relationship between$B(x)$and $C(x)$ if $A(x)$ is always false from the above 2 bullets.

- $A \sqsubseteq C$
- $A \sqsubseteq \forall R.(B \sqcap C)$

orrect Answer

- $A \sqsubseteq \exists R.(B \sqcap C)$

Question 3

0 / 1 pts

Consider the following ALC role assertions.

teaches(Joe, KR)

teaches(Joe, AI)

≤0 teaches(Joe)

Which option is correct?

ou Answered

- It is a TBox and is satisfiable.

As mentioned in the question, these 3 statements are role assertions. An Abox is a set of concept assertions and role assertions. A TBox is a set of terminological axioms. Thus it is an Abox. Note that "≤0 teaches" is a concept, and "≤0 teaches(Joe)" reads "Joe teaches less or equal to 0 things".

- It is an ABox and is satisfiable.
- It is a TBox and is not satisfiable.

orrect Answer

- It is an ABox and is not satisfiable.

Question 4

0 / 1 pts

Which kind of reasoning in ontologies should you use in answering the previous question (question 3)?

- Answering concept queries

ou Answered

- Taxonomies

A Taxonomy is a directed graph where each node is a concept and each directed edge defines the subsumption relationship $\sqsubseteq$ between the 2 nodes of the edge.

Question 3 clearly does not contain a taxonomy and is not asking any question related to a taxonomy.

orrect Answer

- Knowledge Base Satisfiability
- Concept Satisfiability

Question 5

0 / 1 pts

Which option is a suitable ALC concept in description logic for the class of humans “happy pet owner”, using the following concept names and role name?

concept names: Person, Happy, Animal

role name: own

Happy ⊓ Person ⊓ owns.Animal

ou Answered

Person ⊓ Happy ⊓ Animal ⊓ owns(Person, Animal)

There are 2 mistakes in this options.

First, the concept we want to represent must be a kind of person, thus we should not include $Animal$ in this concept.

Second, this option as a whole is not even a valid concept since $owns(Person, Animal)$ is not a concept according to the definition of concept described in Module "Introduction to Description Logics". Please refer to the definition and try to think of a correct way to write this part.

Happy ⊓ Person ⊓ owns(Person, Animal)

orrect Answer

Person ⊓ Happy ⊓ ∃owns.Animal

Question 6

0 / 1 pts

Which option about the use of ontologies in industrial applications is correct?

ou Answered

The nodes in a knowledge graph captures entities, attributes, and relationships.

This option is incorrect. A knowledge graph as a whole captures entities, attributes, and relationships. The nodes only represent entities, while they are labeled with attributes. Typed edges between two nodes capture a relationship between entities.

- The axioms in CYC are automatically generated from web sources.
- The axims in WordNet are written in First-Order Logic.

orrect Answer

- DBpedia automatically evolves as Wikipedia changes.

Question 7

1 / 1 pts

Which option is correct about RDF?

Correct!

- We can use reification to turn a sequence of information into statements in RDF.

This option is correct and an example is shown in Module "Resource Description Framework".

- Every statement has an Internationalized Resource Identifier.
- User defines class hierarchies in RDF to disallow nonsensical statements.
- Resources can be thought of as objects, thus is disjoint from properties.

Question 8

0 / 1 pts

Using the following concept names and role names, which option can correctly capture the knowledge “cars are exactly those vehicles that have wheels and are powered by an engine”?

concept names: Vehicle, Car, Wheel, Engine

role names: hasPart, poweredBy

ou Answered

- Car ⇔ Vehicle ⊓ hasPart.Wheel ⊓ poweredBy.Engine

  This option is syntactically wrong since there is no such symbol ⇔ in ACL.

- Car ⊑ Vehicle ⊓ ∃hasPart.Wheel ⊓ ∀poweredBy.Engine

orrect Answer

- Car ≡ Vehicle ⊓ ∃hasPart.Wheel ⊓ ∃poweredBy.Engine
- Car ⊒ Vehicle ⊓ ∀hasPart.Wheel ⊓ ∀poweredBy.Engine

Question 9

1 / 1 pts

Using the following concept names and role names, which option can correctly capture the knowledge “bicycles have exactly two wheels”?

concept names: Bicycle, Wheel

role names: hasPart

Correct!

- Bicycle ⊑ (=2 hasPart.Wheel)

This option is correct.

First, the English sentence is not saying everything that has exactly 2 wheels must be bicycles, thus we need to write a concept inclusion of the form $C \sqsubseteq D$ instead of a concept definition of the form $C \equiv D$.

Second, we need to use a number restriction as mentioned in Module "Beyond ALC". Here, $(= 2\ hasPart.Wheel)$is a shorthand for $(\geq 2\ hasPart.Wheel) \sqcap (\leq 2\ hasPart.Wheel)$.

Bicycle ≡ (=2 hasPart.Wheel)

Bicycle ⊑ ∃hasPart.Wheel ⊓ ∃hasPart.Wheel

Bicycle ≡ ∀hasPart.Wheel ⊓ (=2 hasPart.Wheel)

Question 10

0 / 1 pts

Which of the following statements is correct?

ou Answered

A ⊓ ∃r.(B ⊔ C) is subsumed by A ⊓ ∃r.B

We can check "X is subsumed by Y" by thinking whether every element in concept X must belong to concept Y. This option is incorrect since it's possible that, say a belongs to $A \sqcap \exists r.(B \sqcup C)$, there exists a b such that b belongs to C but doe not belong to B. In this way, a belongs to $A \sqcap \exists r.(B \sqcup C)$ but does not belong to $A \sqcap \exists r.B$.

A ⊓ ∃r.B is subsumed by A ⊓ ∀r.B

orrect Answer

A ⊓ ∃r.A ⊓ ∀r.B is subsumed by A ⊓ ∃r.B

A ⊓ ∀r.B is subsumed by A ⊓ ∃r.B

FinalExam_4/28/21

Due Apr 28 at 7:05pm **Points** 15 **Questions** 15
Available Apr 28 at 6:10pm - Apr 28 at 7:05pm about 1 hour
Time Limit 55 Minutes

Instructions

- **Your camera must be ON until the end of the exam. Please turn off your phone and show your ID (student ID or passport or Driving license) at the beginning of the exam. OFF camera means that you are absent, and your grade is zero.**

This quiz was locked Apr 28 at 7:05pm.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	50 minutes	15 out of 15

Score for this quiz: **15** out of 15
Submitted Apr 28 at 7:02pm
This attempt took 50 minutes.

Question 1

1 / 1 pts

Is the following statement true or false?

A transition system is a directed graph whose vertices correspond to the states of the world and whose edges are labelled by actions.

Correct!

- (selected) True
- False

Question 2

1 / 1 pts

Is the following statement true or false?

The action “hold” in the following Clingo program is non-deterministic.

```
% File: pendulum.lp
% sorts and object declaration
boolean(t;f).
% effects of hold
right(T+1) :- hold(T), right(T).
left(T+1) :- hold(T), left(T).
% by default, pendulum changes the position
{left(T+1)} :- right(T), T=0..m-1.
{right(T+1)} :- left(T), T=0..m-1.
% fluents are exogenous initially
1{right(0);left(0)}1.
% uniqueness and existence of values for fluents
:- not 1{right(T);left(T)}1, T=1..m.
% exogenous action
{hold(T)} :- T=0..m-1.
```

- True

Correct!

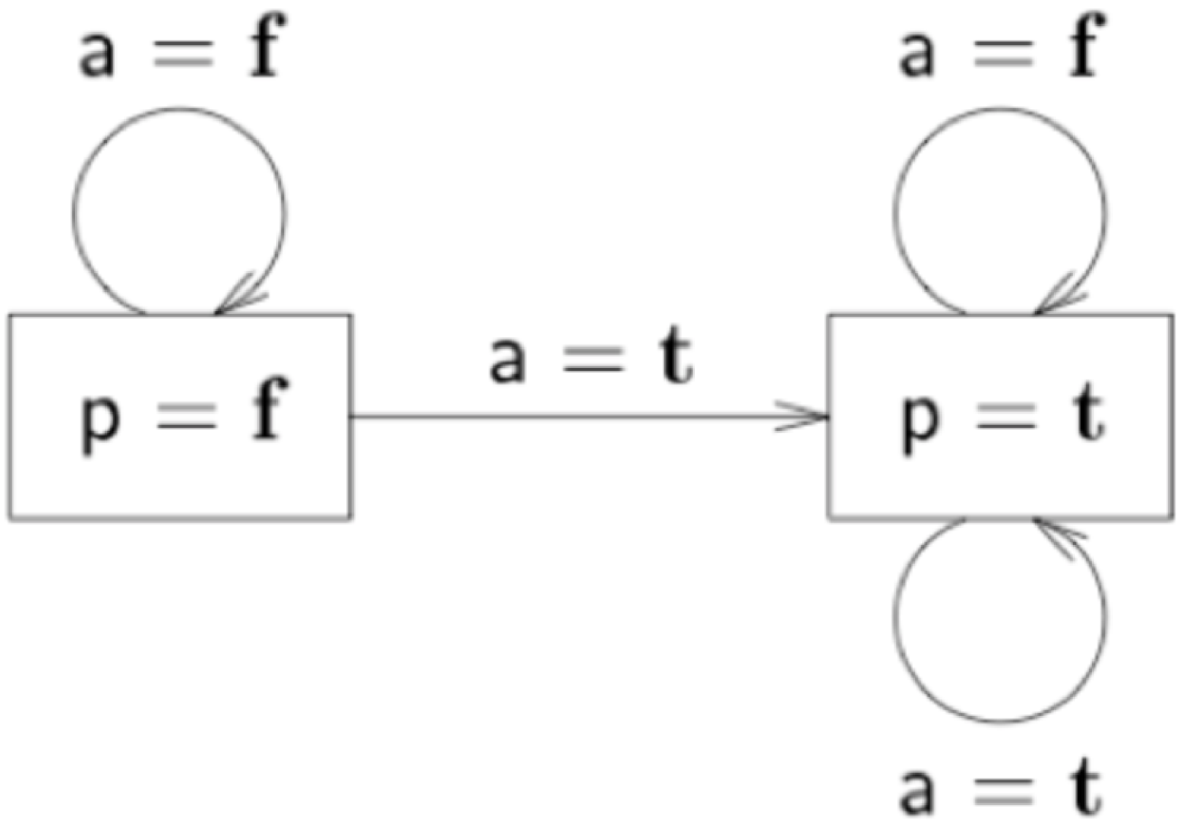
- False

Question 3

1 / 1 pts

Consider the simple transition system in the following image.

Which answer option represents “initial state is exogenous”?



- {a(0)}

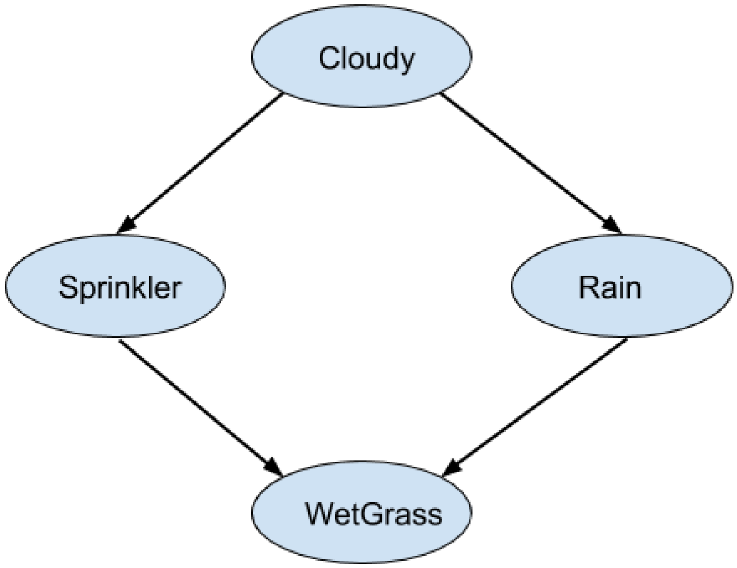
Correct!

- 1{p(t,0); p(f,0)}1
- p(t,1) :- a(0)
- {p(t,1)} :- p(t,0)

Question 4

1 / 1 pts

Given this Bayesian network, how many probability parameters do we need to specify for this network?



Correct!

- (•) 9
- ( ) 8
- ( ) 7
- ( ) 5

Question 5

1 / 1 pts

Consider the following **MLN** with two object constants alice and bob.

Which answer option is the unnormalized probability measure (i.e., weight) of the interpretation {influence(alice, bob)} ? (x, y ranges over both object constants) ?

$2 : smoke(y) \leftarrow smoke(x) \wedge influence(x, y)$

$1 : smoke(alice)$

HINT:

We are considering the **weight** under **Markov Logic Network**.

- exp(4)
- exp(0)
- Correct! exp(8)
- exp(2)

Question 6

1 / 1 pts

In ALC, which category does “Student(John)” belong to?

- Tbox
- Correct! Concept assertion
- Role assertion
- Concept inclusion

Question 7

1 / 1 pts

In regard to syntax relationship, which of the following statements is correct?

1 ○ (>=2 RC) in description logics can be written as R min 2 in OWL.

2 ○ ∀ R.C in description logics can be written as R all C in OWL.

3 ○ ∃ R.C in description logics can be written as R some C in OWL.

4 ○ C1 ⊑ ¬ C2 can be written as C1 SubClassOf C2 in OWL.

○ 2

Correct!

◉ 3

○ 4

○ 1

Question 8

1 / 1 pts

The completion of an ASP program Π is defined as the union of Π and

$$A \rightarrow \bigvee_{A \leftarrow Body \in \Pi} Body$$

for each ground atom A.

Given ASP program Π:

$p \leftarrow \neg q$

$q \leftarrow \neg r$

What is the formula after the completion?

1 ○ $p \leftrightarrow \neg q$
r

2 ○ $p \leftrightarrow \neg q$
$q \leftrightarrow \neg r$
$r \leftarrow \bot$

3 ○ $p \leftrightarrow \neg q$
$q \leftrightarrow \neg r$
$\neg r$

4 ○ $p \leftrightarrow \neg q$
$q \leftrightarrow \neg p$

- 2

Correct!

- 3

- 1

- 4

Question 9

1 / 1 pts

Examine the following probability distribution table.

Weather	Temperature	$\mu(w)$
sunny	hot	0.10
sunny	mild	0.20
sunny	cold	0.10
cloudy	hot	0.05
cloudy	mild	0.35
cloudy	cold	0.20

What is the probability that the weather is sunny given that the temperature is hot?

P(Weather = sunny | Temperature = hot) = ?

- 0.20

Correct!

- 0.67
- 0.25
- 0.10

Question 10

1 / 1 pts

Which of the following options is the transformation of π_x(Man(a) ∧ ∃ hasChild.Female)?

1 ○ Man(a) ∧ ∃ y (hasChild(x, y) ∧ female(y)).

2 ○ Man(a) ∧ ∃ x (hasChild(x, y) ∧ female(y)).

3 ○ ∀ a Man(a) ∧ ∃ y (hasChild(x, y) ∧ female(y)).

4 ○ ∀ x Man(x) ∧ ∃ y (hasChild(x, y) ∧ female(y)).

○ 3

○ 4

○ 2

Correct! ◉ 1

Question 11

1 / 1 pts

What will be the stable model if p is true at time 0 and a(0) is true in the following program?

p(t, 1) :- a(0).

1{p(t, 0); p(f, 0)}1.

{a(0)}.

{p(t, 1)} :- p(t, 0).

{p(f, 1)} :- p(f, 0).

:- not 1{p(t, 1); p(f, 1)}1.

- {p(t, 0), a(0), p(f, 1)}
- {p(t, 1)}
- {p(t, 0), a(0), p(t, 1), p(f, 1)}
- Correct! {p(t, 0), a(0), p(t, 1)}

Question 12

1 / 1 pts

Consider the following clingo program with 5 rules.

bird(X) :- residentBird(X).

bird(X) :- migratoryBird(X).

:- residentBird(X), migratoryBird(X).

residentBird(jo).

migratoryBird(jo).

What are the stable models?

- {bird(jo), residentBird(jo), migratoryBird(jo)}

Correct!
- Unsatisfiable
- { residentBird(jo), migratoryBird(jo)}
- { bird(jo)}

Question 13

1 / 1 pts

Which rule is **not** allowed in clingo given the following rule?

boolean(true; false).

Correct!
- 1{hasBananas(BB, 0)}1.
- 1{hasBananas(BB, 0): boolean(BB)}1.
- :-not 1{hasBananas(BB, T): boolean(BB)}1, T=1..m
- :- not 1{hasBananas(BB, T)}1, T=1..m.

Question 14

1 / 1 pts

Which statement of concept satisfiability is correct?

1 ○ A concept C is satisfiable with respect to a knowledge base K even if K itself is not satisfiable.

2 ○ A concept C is satisfiable with respect to a knowledge base K if there exists a model I of K such that $C^I \neq \emptyset$.

3 ○ ParentOfOnlyMaleChildren ≡ Person ⊓ ∀ hasChild.Male.

4 ○ Consider the knowledge base K below.

Male ⊑ ¬Female

The concept

Male ⊓ ¬Female

is satisfiable with respect to K.

Correct!

- 2
- 3
- 1
- 4

Question 15

1 / 1 pts

In the Pendulum problem, which of the following options represents that the position of pendulum is a non-inertial fluent?

1 ○ {right(T+1)} :- right(T), T=0..m-1.

2 ○ right(T+1) :- hold(T), right(T).

3 ○ {left(T+1)} :- right(T), T=0..m-1.

4 ○ {left(T+1)} :- left(T), T=0..m-1.

- ○ 2
- ○ 1
- ◉ 3 Correct!
- ○ 4

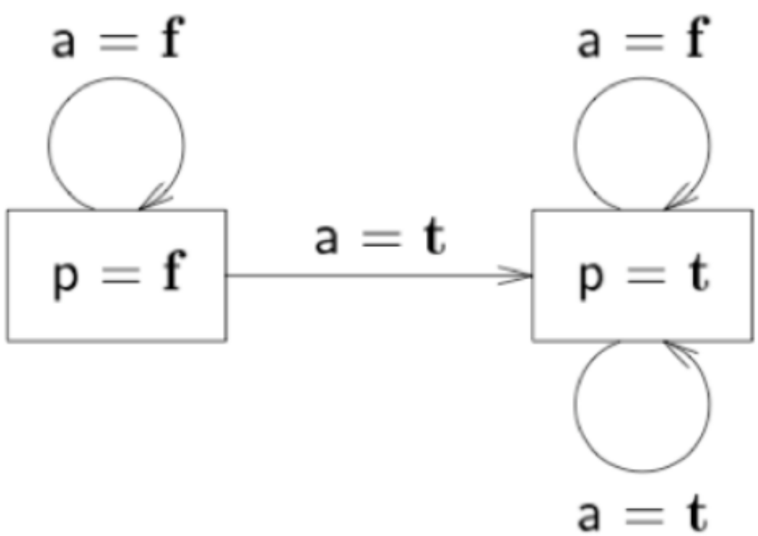
Quiz Score: **15** out of 15

Final Exam Practice Questions

1. Transition Systems: Directed graphs whose vertices correspond to the actions and edges are labeled by states.

   Is the definition of Transition Systems true or false?

2. Which are the stable models for the given transition system?



   a. { p(f, 0), p(t, 1)}
   b. { p(f, 0), a(0)}
   c. { p(t, 0), p(t, 1), a(0)}
   d. { p(f, 0), p(f, 1)}

3. Choose all the fluents from the given options.
   a. Monkey has the banana
   b. Monkey grasps the banana
   c. The location of the box
   d. Monkey is on the box

4. For the state, given above, choose the correct common-sense law of intertia.
    a. { p(t, 1) } :- p(f, 0).

       { p(f, 1) } :- p(f, 0).

       :- not 1 { p(t, 1); p(f, 1) } 1.
    b. { p(f, 1) } :- p(f, 0).

       { p(f, 1) } :- p(t, 0).

       :- not 1 { p(t, 1); p(f, 1) } 1.
    c. { p(t, 1) } :- p(t, 0).

       { p(f, 1) } :- p(f, 0).

       :- not 1 { p(t, 1); p(f, 1) } 1.

5. Which option is the most probable stable model of the following LPMLN Program?

   1: p

   α : p <- q

   α : ⊥ <- ~q

    a. Ø
    b. {p}
    c. {q}
    d. {p. q}

6. If A and B are random variables, which option about probability is impossible?
    a. P(A | B) = P(A, B)
    b. P(B | A) > P(A | B)
    c. P(A | B) < P(A, B)
    d. P(B) > P(A, B)

7. Consider the given full-joint distribution for Boolean variables A, B, and C. What is the value of P(A = 1 | B = 1, C = 0)?

A	B	C	P(A,B,C)
0	0	0	0.2
0	0	1	0.1
0	1	0	0.15
0	1	1	0.05
1	0	0	0.05
1	0	1	0.25
1	1	0	0.05
1	1	1	0.15

a. 0.2
b. 0.25
c. 0.5
d. 0.75

8. Which option about the use of ontologies in industrial applications is correct?
a. The axioms in WordNet are written in FOL.
b. The axioms in CYC are automatically generated from web sources.
c. DBpedia automatically evolves as Wikipedia changes.
d. The nodes in a knowledge graph captures entities, attributes, and relationships

9. Which is correct regarding the Resource Description Framework?
a. User defines class hierarchies in RDF to disallow nonsensical statements.
b. We can reification to turn a sequence of information into statements in RDF.
c. Every statement has an IRI (Internationalized Resource Identifier).
d. Resources can be thought of as objects.

10. Which option is correct?
a. $A \sqcap \exists r.(B \sqcup C)$ is subsumed by $A \sqcap \exists r.B$
b. $A \sqcap \exists r.B$ is subsumed by $A \sqcap \forall r.B$
c. $A \sqcap \exists r.A \sqcap \forall r.B$ is subsumed by $A \sqcap \exists r.B$
d. $A \sqcap \forall r.B$ is subsumed by $A \sqcap \exists r.B$

Practice Questions for Final (Set I)

1. Consider the scenario where two agents lift the opposite ends of a table upon which various objects have been placed. If one end of the table has been raised, the objects on the table will fall off. But if both ends are lifted simultaneously, the objects on the table will remain fixed. The clingo program for this scenario is as follows:

```
boolean(t;f).

end(leftEnd;rightEnd).

height(low;high).

onTable(f,T) :- level(leftEnd,H,T), level(rightEnd,H1,T), H!=H1.

level(E,high,T+1) :- lift(E,T).

:- lift(E,T), level(E,high,T).

1{level(E,HH,0): height(HH)}1 :- end(E).

1{onTable(BB,0): boolean(BB)}1.

:- not 1{level(E,HH,T)}1, end(E), T=1..m.

:- not 1{onTable(BB,T)}1, T=1..m.

{lift(E,T)} :- end(E), T=0..m-1.

{level(E,H,T+1)} :- level(E,H,T), T=0..m-1.

{onTable(B,T+1)} :- onTable(B,T), T=0..m-1.
```

Which of the following options represents the statement "actions are exogeneous"?

A. {lift(E,T)} :- end(E), T=0..m-1.
B. level(E,high,T+1) :- lift(E,T).
C. {level(E,H,T+1)} :- level(E,H,T), T=0..m-1.
D. :- lift(E,T), level(E,high,T).

---

2. Consider the Monkey and Bananas problem represented by the following clingo program.

```
%% sort and object declaration

boolean(t;f).

object(monkey;bananas;box).

location(l1;l2;l3).

%% state constraints

loc(bananas,L,T) :- hasBananas(t,T), loc(monkey,L,T).

loc(monkey,L,T) :- onBox(t,T), loc(box,L,T).
```

```
%% effect and preconditions of actions

% walk

loc(monkey,L,T+1) :- walk(L,T).

:- walk(L,T), loc(monkey,L,T).

:- walk(L,T), onBox(t,T).

% pushBox

loc(box,L,T+1) :- pushBox(L,T).

loc(monkey,L,T+1) :- pushBox(L,T).

:- pushBox(L,T), loc(monkey,L,T).

:- pushBox(L,T), onBox(t,T).

:- pushBox(L,T), loc(monkey,L1,T), loc(box,L2,T), L1 != L2.

% climbOn

onBox(t,T+1) :- climbOn(T).

:- climbOn(T), onBox(t,T).

:- climbOn(T), loc(monkey,L1,T), loc(box,L2,T), L1 != L2.

% climbOff

onBox(f,T+1) :- climbOff(T).

:- climbOff(T), onBox(f,T).

% graspBananas

hasBananas(t,T+1) :- graspBananas(T).

:- graspBananas(T), hasBananas(t,T).

:- graspBananas(T), onBox(f,T).

:- graspBananas(T), loc(monkey,L1,T), loc(bananas,L2,T), L1 != L2.

% disallow concurrent actions

:- walk(L,T), pushBox(L,T).

:- walk(L,T), climbOn(T).

:- pushBox(L,T), climbOn(T).

:- climbOff(T), graspBananas(T).

%% domain independent axioms

% fluents are initially exogenous

1{hasBananas(BB,0):boolean(BB)}1.
```

```
1{onBox(BB,0):boolean(BB)}1.

1{loc(O,LL,0):location(LL)}1 :- object(O).

% uniqueness and existence of fluent values

:- not 1{loc(O,LL,T)}1, object(O), T = 1..m.

:- not 1{onBox(BB,T)}1, T = 1..m.

:- not 1{hasBananas(BB,T)}1, T = 1..m.

% actions are exogenous

{walk(L,T)} :- location(L), T = 0..m-1.

{pushBox(L,T)} :- location(L), T = 0..m-1.

{climbOn(T)} :- T = 0..m-1.

{climbOff(T)} :- T = 0..m-1.

{graspBananas(T)} :- T = 0..m-1.

% commonsense law of inertia

{hasBananas(B,T+1)} :- hasBananas(B,T), T=0..m-1.

{onBox(B,T+1)} :- onBox(B,T), T=0..m-1.

{loc(O,L,T+1)} :- loc(O,L,T), T=0..m-1.
```

Which of the following rules represent the post-diction query: "The monkey walked to location L3 and then pushed the box. Does it follow that the box was initially at L3?"

A. :- not walk(l3,0).
   :- not 1{pushBox(LL,1):location(LL)}.
   :- loc(box,l3,0).

B. :- not loc(monkey,l1,0).
   :- not loc(bananas,l2,0).
   :- not loc(box,l3,0).
   :- not walk(l3,0).
   :- not pushBox(l2,1).
   :- #count{LL:loc(monkey,LL,2), loc(bananas,LL,2), loc(box,LL,2)}=1.

C. :- not loc(monkey,l1,0).
   :- not loc(bananas,l2,0).
   :- not loc(box,l3,0).
   :- not hasBananas(f,0).
   :- not hasBananas(t,m).

3. Consider a simple transition system represented by the following clingo program.

```
% File ‘simple.lp’

boolean(t;f).

% direct effect

p(t,T+1) :- a(T), T=0.

% initial status are exogenous

1{p(B,0):boolean(B)}1.

% uniqueness and existence of values

1{p(B,T) : boolean(B)}1 :- T=1.

% actions are exogenous

{a(T)} :- T=0.

% commonsense law of inertia

{p(B,T+1)} :- p(B,T), T=0.

#show p/2.
```

How many stable models does this program have?

A. 2
B. 4
C. 6
D. 8

---

4. In the semantics of LP^{MLN}, the following formulae represent __________-based weight.

$$W_\Pi(I) = e^{-\sum_{w:R\in\Pi, I\not\models R} w}$$

$$P_\Pi = \lim_{\alpha\to\infty}\left(\frac{W_\Pi(I)}{\sum_J W_\Pi(J)}\right)$$

A. Penalty
B. Reward

---

5. Consider the following full joint distribution for Boolean variables A, B, and C.

A	**B**	**C**	**P (A, B, C)**
0	0	0	0.1
0	0	1	0.2
0	1	0	0.05
0	1	1	0.15
1	0	0	0.2
1	0	1	0.1
1	1	0	0.15
1	1	1	0.05

Which of the following is closest to the value of P (A=0|B=1, C=1)?

A. 0.75
B. 0.25
C. 0.5
D. 0.33

---

6. Consider the following LP^{MLN} program Π.

<u>KB1</u>

$\alpha: B(x) \leftarrow RB(x)$

$\alpha: B(x) \leftarrow MB(x)$

$\alpha: \leftarrow RB(x), MB(x)$

<u>KB2</u>

$\alpha: RB(birb)$

<u>KB3</u>

$\alpha: MB(birb)$

What is $W_{\Pi}(\{M(birb)\})$?

A. $e^{2\alpha}$
B. $e^{3\alpha}$
C. $e^{4\alpha}$
D. 0

---

7. Which of the following are true about very large ontologies?

A. DBPedia is written in RDF.
B. Cyc is written in CycL, which is an extension of Predicate Logic.
C. WordNet is a large lexical database of English words.
D. YAGO2 incorporates temporal and spatial data.

---

8. What are the basic ideas of RDF?

A. Database, Relations, Entities
B. Description, Framework, Resources
C. Subject, Object, Predicate
D. Resources, Properties, Statements

---

9. Which of the following are examples of Concept Inclusions?

A. Disjointness of Concepts
B. Coverings
C. Domain Restrictions
D. Range Restrictions

---

10. Using the following concept and role names, which option correctly captures the knowledge "Latte is exactly that Coffee Drink which has Espresso and Steamed Milk as its ingredients".

Concept Names: Latte, CoffeeDrink, Espresso, SteamedMilk

Role Names: hasIngredient

A. "$Latte \equiv CoffeeDrink \sqcap \exists hasIngredient.Espresso \sqcap \exists hasIngredient.SteamedMilk$"
B. "$Latte \sqsupseteq CoffeeDrink \sqcap \exists hasIngredient.Espresso \sqcap \exists hasIngredient.SteamedMilk$"
C. "$Latte \sqsubseteq CoffeeDrink \sqcap \exists hasIngredient.Espresso \sqcap \exists hasIngredient.SteamedMilk$"
D. "$Latte \equiv CoffeeDrink \sqcap \forall hasIngredient.Espresso \sqcap \forall hasIngredient.SteamedMilk$"

CSE579: Knowledge Representation and Reasoning

Solutions to Practice Questions for Final (Set I)

1. A

```
% actions are exogenous
{lift(E,T)} :- end(E), T=0..m-1.
```

2. A

```
% File 'monkey-postdiction.lp'
:- not walk(l3,0).
:- not 1{pushBox(LL,1):location(LL)}.
:- loc(box,l3,0).
```

3. C

```
clingo version 5.5.0
Reading from stdin
Solving...
Answer: 1
p(t,0) p(f,1)
Answer: 2
p(f,1) p(f,0)
Answer: 3
p(t,1) p(t,0)
Answer: 4
p(t,1) p(t,0)
Answer: 5
p(t,1) p(f,0)
Answer: 6
p(t,1) p(f,0)
SATISFIABLE
```

4. A

The formulae represent Penalty-based weight.

---

5. A

$P(A = 0|B = 1, C = 1)$

$= P(A = 0, B = 1, C = 1)/P(B = 1, C = 1)$

$= (0.15)/(0.15 + 0.05)$

$= 0.15/0.20$

$= 0.75$

---

6. B

The set {MB(birb)} satisfies the following rules:

α: $B(x) \leftarrow RB(x)$

α: $\leftarrow RB(x), MB(x)$

α: $MB(birb)$

Hence,

$$W_{\Pi}(\{\text{MB(birb)}\}) = e^{3\alpha}$$

---

7. A, C, D

CycL is an extension of First-Order Logic.

---

8. D

The basic ideas of RDF are Resources, Properties, and Statements.

---

9. A, B, C, D

Disjointness of Concepts, Coverings, Domain Restrictions, and Range Restrictions are all examples of Concept Inclusions.

---

10. A

A. Correct Statement

B. $\sqsupseteq$ is incorrect, $\equiv$ is used to capture knowledge

C. $\sqsubseteq$ is incorrect, $\equiv$ is used to capture knowledge

D. $\forall$ is incorrect when referring to multiple ingredients

---