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