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

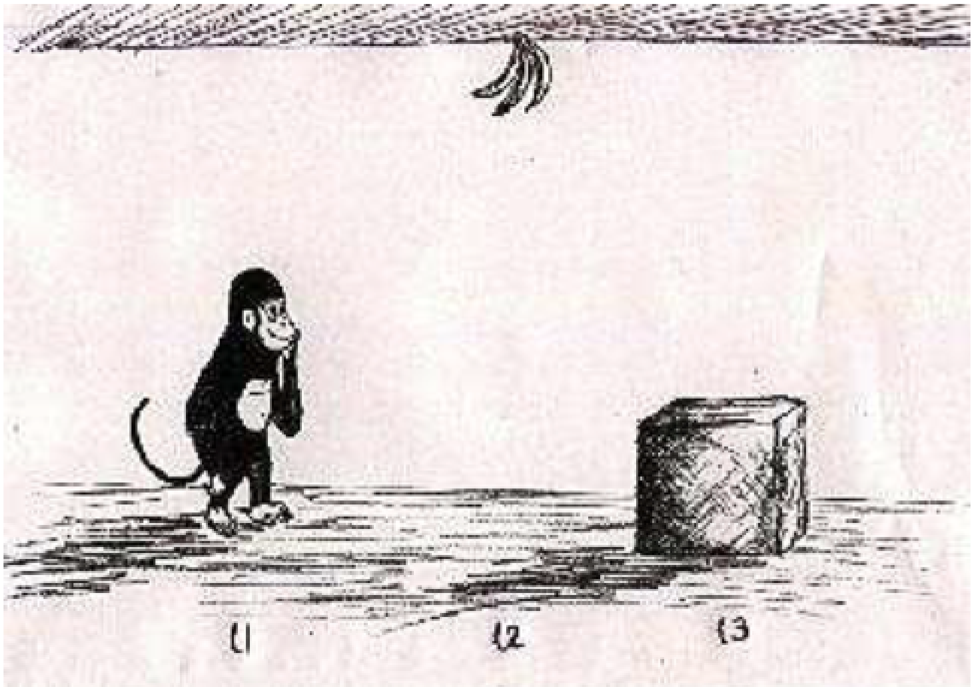
	Attempt	Time	Score
LATEST	Attempt 1	8 minutes	10 out of 10

Score for this attempt: **10** out of 10
Submitted Mar 26 at 4:53pm
This attempt took 8 minutes.

Question 1

1 / 1 pts

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



- The jump of the monkey.
- Monkey moves the box.
- Monkey grasps the bananas.

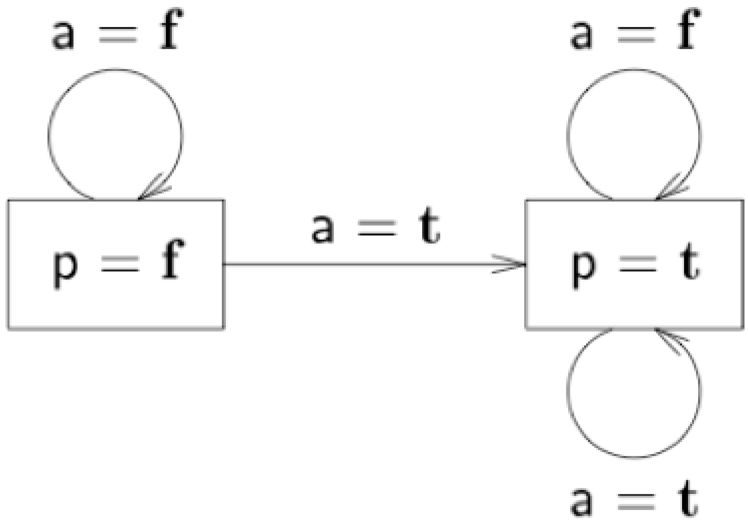
Correct!

- The location of the box.

Question 2

1 / 1 pts

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



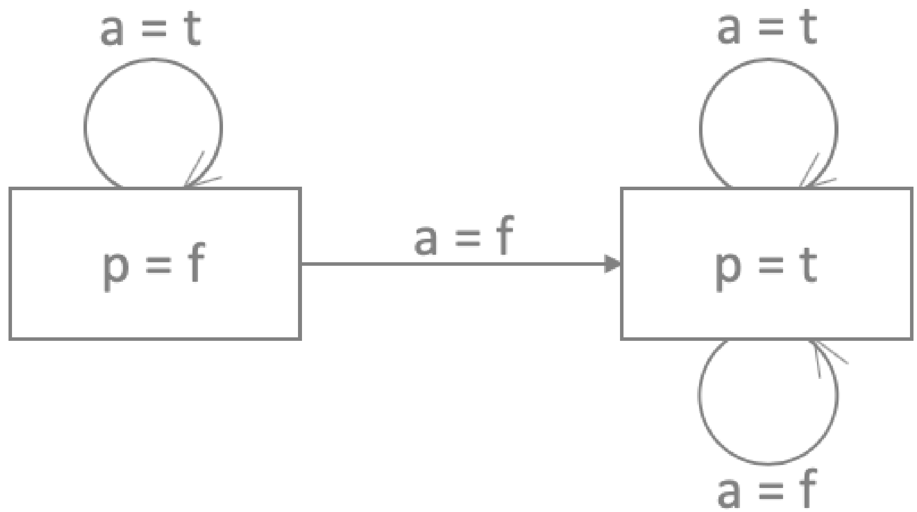
- {p(t,1)} :- p(t,0).
- 1{p(t,0); p(f,0)}1.
- Correct! {a(0)}.
- p(t,1) :- a(0).

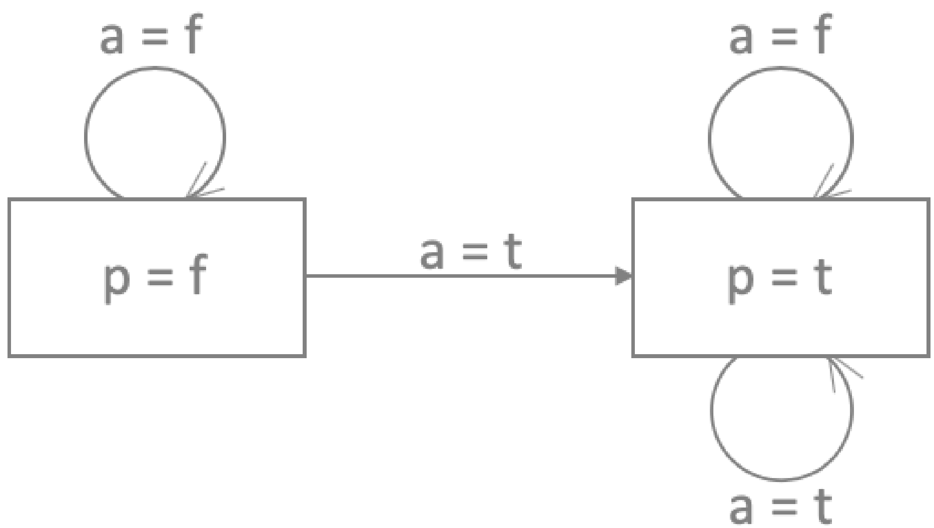
Question 3

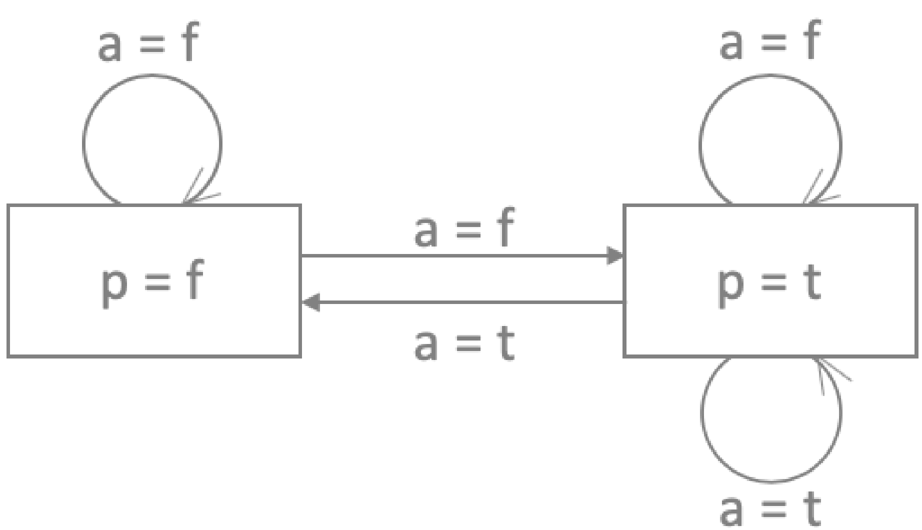
1 / 1 pts

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.)

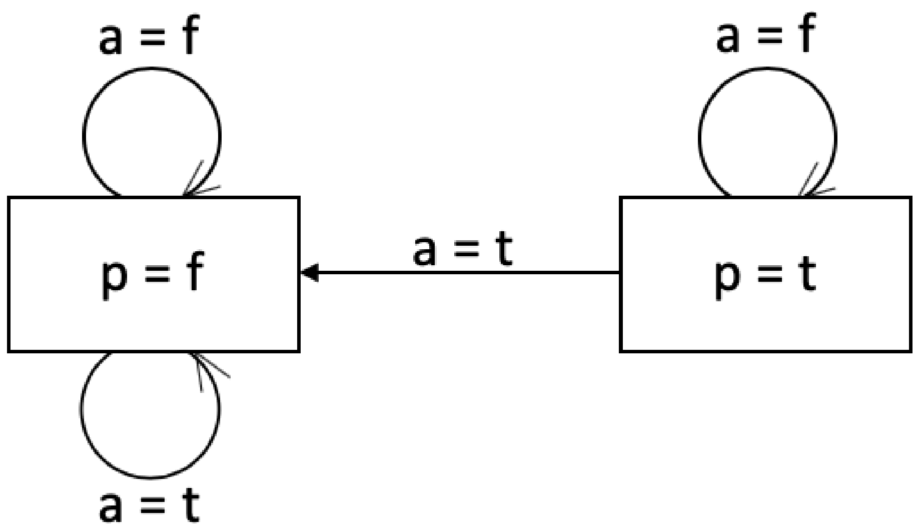
```
boolean(t;f).
p(f, T+1) :- a(T), T=0..m-1.
1{p(B,0): boolean(B)}1.
:- T=1..m, not 1{p(B, T): boolean(B)}1.
{a(T)} :- T=0..m-1.
{p(B, T+1)} :- p(B, T), T=0..m-1.
```


a = t
a = t
p = f
a = f
p = t
a = f


a = f
a = f
p = f
a = t
p = t
a = t


a = f
a = f
p = f
a = f
a = t
p = t
a = t

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
- 2
- 6
- 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.
```

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

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

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

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

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

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.

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?

Correct!

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

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

---

loc(monkey, l1, 0).

loc(bananas, l1, 0).

loc(box, l2, 0).

hasBananas(f, 0; t, m).

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

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

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

:- not hasBananas(f, 0).

:- not hasBananas(t, 1).

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

hasBananas(f, 0).

hasBananas(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!

- (•) 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?

- 1
- 0
- 4
- 2

Correct!

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
- 4
- 0

Correct!

- 1

Question 10

1 / 1 pts

Which option is correct about action description languages?

Correct!

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

Quiz Score: **10** out of 10