Chapter 8

Question 2 (a)

For creating the program, I followed the structure of blocks world program and section 8.4.2, where encoding a program is given. First I created the AL system provided in the question, which says that *a causes f if g* and *h if f,g.* Then I created the inertia. Integrating these with CWA for fluents and actions. I also gave the actions for testing and created a specific initial state for testing purposes.

The program created for this part is –

```
%given in question
% a causes f if g
% h if f,g

% fluents from question
fluent(inertial,f).
fluent(inertial,g).
fluent(defined, h).

% actions
action(a).

%Steps
#const n=1.
step(0,n).

% laws defined as per provided rules in question
holds(f, I+1) :- occurs(a, I),
                 holds(g, I),
                 I < n.
holds(h, I) :-  holds(f, I), holds(g, I).

% CWA for fluents
-holds(F,I) :- fluent(defined,F),
               step(I),
               not holds(F,I).

% CWA for actions
-occurs(A,I) :- action(A), step(I),
                not occurs(A,I).

% Inertia:
holds(F,I+1) :- fluent(inertial,F),
```

```
            holds(F,I),
            not -holds(F,I+1), I < n.
-holds(F,I+1) :- fluent(inertial,F),
             -holds(F,I),
             not holds(F,I+1),
             I < n.

%% Specific actions for testing.
% action provided
%occurs(a,0).

%% Change these at will:
% initial state.
%holds(f,0).
%holds(g,0).
%holds(h,0).

% effect of action = new state
%#show holds/2.
```

Question 2(b)

Only the fourth part, i.e., {f, g,-h} are invalid state as per the created program. For this, I used the logic that for checking a state to be valid, then the state should be the answer set of *πc(SD) U {f,g}.*

I – valid

II - valid

III – valid

IV - invalid