Problem:

Two airplanes are moving toward an airport and wish to land when they arrive. The airport has airspace restrictions that approaching planes must arrive at a certain circling distance ($d_c$) and check if another plane is already landing. If so, they must wait a certain length of time ($t_c$) near the airport and then fly off if landing permission is not given. Otherwise, they may land as soon as the runway is clear. Any airplane requires 1 time unit to land. The airport is interested in avoiding any situation in which an airplane waits so long that it gives up and flies off from the airport. For the situation in which a small airplane (capable of speeds between 4 and 6) and a large airplane (capable of speeds between 15 and 18) approach the airport from a distance of 50 away, what circling distance ($d_c$) and circling time ($t_c$) restrictions will allow both airplanes to land on their first attempt? Also, if both airplanes were in an emergency situation and thus needed to land immediately and had no way to know if another plane was landing, determine at what distance the planes should initiate landing and how quickly they would reach the ground. (Hint: Do the second part by eliminating the Flying Out state and the landing counter ($k$) and adding a parameter for landing time). The automaton for the general airplane is below.

Automaton:
Solution:

(a) Automata:

```
var
x,y,s,t: analog;
k: discrete;
circle_dist,circle_time: parameter;

automaton airplane1
synclabs : ;
initially flying_in & x=60 & s=0;

loc flying_in: while x>=circle_dist wait {dx in [-6,-4],ds=0}
    when x=circle_dist & k=2 do {s'=0} goto circling;
    when x=circle_dist & k=0 do {s'=0,k'=1} goto landing;
loc circling: while s<=circle_time wait {dx=0,ds=1}
    when k=0 do {s'=0,k'=1} goto landing;
    when s=circle_time & k=2 do {s'=0} goto flying_out;
loc flying_out: while x<=60 wait {dx=3,ds=0}
    when x=60 goto flying_in;
loc landing: while s<=1 wait {dx=0,ds=1}
    when s=1 do {s'=0,k'=0} goto grounded;
loc grounded: while True wait {dx=0,ds=0}
end

automaton airplane2
synclabs : ;
initially flying_in & y=60 & t=0;
loc flying_in: while y>=circle_dist wait {dy in [-18,-15],dt=0}
    when y=circle_dist & k=1 do {t'=0} goto circling;
    when y=circle_dist & k=0 do {t'=0,k'=2} goto landing;
loc circling: while t<=circle_time wait {dy=0,dt=1}
    when k=0 do {t'=0,k'=2} goto landing;
    when t=circle_time & k=1 do {t'=0} goto flying_out;
loc flying_out: while y<=60 wait {dy=4,dt=0}
    when y=60 goto flying_in;
loc landing: while t<=1 wait {dy=0,dt=1}
    when t=1 do {t'=0,k'=0} goto grounded;
loc grounded: while True wait {dy=0,dt=0}
end
```

Analysis:

```
var init_reg, final_reg, reached: region;
init_reg := loc[airplane1]=flying_in & loc[airplane2]=flying_in & x=50 & y=50 & s=0 &
             t=0 & k=0;
final_reg := loc[airplane1]=flying_out | loc[airplane2]=flying_out;
reached := reach forward from init_reg endreach;
prints "Conditions under which system violates safety requirement":
print omit all locations
    hide non_parameters in reached & final_reg endhide;

Conditions (in which a plane makes more than one attempt at landing):
\[ d_c \geq 10 \ t_c + 40, \ d_c \leq 50, \ t_c \geq 0 \]
(b) Automata:

```plaintext
var
  x,y,s,t: analog;
  circle_dist, land_time: parameter;

automaton airplane1
  synclabs : ;
  initially flying_in & x=50 & s=0;
  loc flying_in: while x>=circle_dist wait {dx in [-6,-4], ds=0}
      when x=circle_dist do {s'=0} goto landing;
  loc landing: while s<=land_time wait {dx=0, ds=1}
      when s=land_time do {s'=0} goto grounded;
  loc grounded: while True wait {dx=0, ds=0}
end

automaton airplane2
  synclabs : ;
  initially flying_in & y=50 & t=0;
  loc flying_in: while y>=circle_dist wait {dy in [-18,-15], dt=0}
      when y=circle_dist do {t'=0} goto landing;
  loc landing: while t<=land_time wait {dy=0, dt=1}
      when t=land_time do {t'=0} goto grounded;
  loc grounded: while True wait {dy=0, dt=0}
end
```

Analysis:

```plaintext
var init_reg, final_reg, reached: region;

init_reg := loc[airplane1]=flying_in & loc[airplane2]=flying_in & x=50 & y=50 & s=0 &
             t=0;
final_reg := loc[airplane1]=landing & loc[airplane2]=landing;
reached := reach forward from init_reg endreach;
prints "Conditions under which system violates safety requirement";
print omit all locations
  hide non_parameters in reached & final_reg endhide;
```

Conditions (in which both planes try to land at the same time):

\[ d_c \geq 50 - 10 t_L, \quad d_c \leq 50 \]