Assignment 1 - Promela model for TCAS and verification with Spin

Harish Loganathan - A0092721

Assumptions

In addition to the assumptions specified in the project specification document (assignment1.pdf) I have made the following assumptions

1. The only possible RA messages are climb, descend, climb_faster and maintain.
2. The only possible TA message is Traffic. When a TA alert is detected, it is considered just as a warning and the aircraft specified in the model does not act on it.
3. Speed of the aircraft varies from 1 (slowest) to 3 (fastest). Higher speed has a bigger RA and TA region.
4. In the original TCAS system, in order to detect other aircrafts in the protected volume an aircraft will send messages into the airspace at some particular frequency and the other aircrafts respond at a particular frequency. So, to mimic that behaviour, I have created a global buffered channel called **interrogation** into which all aircrafts will send its interrogation messages. An interrogation message is a reference to the locally defined buffered channel **recv_chan**. All aircrafts will nondeterministically choose to read from the interrogation channel and send its current location, direction and velocity to the channel reference received from the interrogation channel. Detection of aircrafts in the RA and TA region will happen only when an aircraft receives a response to its interrogation.

Insights and Difficulties

1. It is apparently not possible to verify properties for my model exhaustively because of the program size. Therefore, I specified the option '-DMEMLIM=2048' (2 gigs) when compiling my verifier program.
2. When I ran the verifier program it complained that the VECTORSZ is too small. To avoid this error I specified '-DVECTORSZ=65536' when compiling my verifier program.
3. I have also used the '-DBITSTATE' option for super trace verification.
4. Although I tried to have it, my model does not have a valid end state. The aircrafts could possibly be in any state when the specified memory limit is reached. I have specified the -E option to ignore invalid end states while running my verifier.
5. In order to force the simulation to run to completion within reasonable time, I allow the aircraft processes to move only about a 100 times before terminating. This change helped me to debug many issues in my model.
6. When an aircraft process terminates, there might be other processes trying to send messages to this aircraft. So I limited the number of messages that an aircraft can have on the interrogation channel to 1 and I handled timeouts in synchronous communications with an explicit timeout guard.
7. I was trying to specify a lot of properties involving <> (eventually) operator. But irrespective of the property, even if I give a condition which cannot be satisfied by my model, the verifier never complained that my property was violated. The reason I think we can attribute to this is that my verifications are not exhaustive. I may be wrong.

8. I realize that there are problems with my approach of sending interrogations and receiving responses. It is possible that an aircraft miss some aircrafts which come into its RA or TA region. I decided to leave it that way because this behavior reflects reality in some sense. I did not want to lookup the airspace for aircrafts around because that would be like RADAR. In the real system, messages may get lost or other aircraft may not be equipped with the TCAS system.

Differences between versions

1. Version 1 satisfies all the requirements specified in the assignment specification including the 'Bonus' where speed is implemented with counters. The boundary of the RA region is (RA_proportionality_const * speed) cells away from the aircraft’s current location. Similarly The boundary of the TA region is (TA_proportionality_const * speed) cells away. Message communication for interrogation and reply is by buffered channels and for collision avoidance maneuvers is via synchronous channels. Every aircraft chooses its initial position in the airspace, its velocity and direction nondeterministically. Also if another aircraft is detected in the RA region, the advisory for collision avoidance is chosen non-deterministically.

2. Version 2 has the following properties specified on the model.

(i) Local assertion which checks the sanity of distance computation

```
assert (! (otherAircraftInRA) || (xDist<=(RA_proportionality_const*velocity) &&
yDist<=(RA_proportionality_const*velocity) &&
zDist<=(RA_proportionality_const*velocity)));

assert (! (otherAircraftInTA) || (xDist<=(TA_proportionality_const*velocity) &&
yDist<=(TA_proportionality_const*velocity) &&
zDist<=(TA_proportionality_const*velocity)));
```

(ii) LTL property which says an aircraft can send a message on the interrogation channel only if it has received replies for all its previously sent messages. **interrogation** channel cannot have more messages than number of aircrafts.

```
ltl num_interrogations { [] (len(interrogation) < NumAircraft) };
```

(iii) a never claim which maintains that an aircraft never moves after collision. Collided aircrafts are assumed to have crashed and removed from the airspace.

```
never {
        do
        ::moveCalledAfterCollision == 1; break;
        ::else
        od;
}
```

Where **moveCalledAfterCollision** is a bit which is set if move() is called after collision.