This assignment can be done in teams of up to 2 people. Each student is responsible for contacting other students and form a team. Individual submissions are accepted but not encouraged. In particular, no reduction of work will be granted to them.

Download the accompanying files `hw1a.als` and `hw1b.als`, type your name(s) and your solutions in that file as indicated there, and submit it on ICON. For team solutions, *only one member of the team should submit*, but make sure to write down the name of both team members in the model files.

The grade for the assignment will be given on an individual basis. For team submissions, both students must also submit an evaluation of how well they and their teammate performed as team members. See Piazza for instructions on that. Each evaluation is confidential and will be incorporated in the calculation of the grade.

*Be sure to review the syllabus for details about this course’s cheating policy.*

**Hint:** Develop and test your Alloy model incrementally by commenting out selected parts of it (such as the assertions already provided for Question 4 in Part B).

### A Encoding more facts in the Family model

Consider the following signatures for a version of the family model seen in class which is provided in `hw1a.als`.

```alloy
abstract sig Person {
    children: set Person,
    likes: set Person,
    loves: set Person,
    spouse: lone Person
}

sig Man, Woman extends Person {}

one sig John extends Man {}

one sig Mary extends Woman {}
```
1. Add a function `wife` that takes as argument a person and returns that person’s wife, if any.

2. Add a function `husband` that takes as argument a person and returns that person’s husband, if any. Define `husband` in terms of `wife`.

3. Add a predicate `isMarried` which applies to one argument of type `Person` and is true if and only if that person has a spouse.

4. Add a predicate `LikedByChildren` which applies to one argument of type `Person` and is true if and only if that person has a child that likes him or her.

5. Formalize and add the following facts to this model. If you find one of the English sentences ambiguous, clarify in an Alloy comment how you are disambiguating it.

Feel free to add more auxiliary functions or predicates if you want, but do not modify the given signatures or add new ones.

If a fact makes a model inconsistent, explain why.

(a) Everybody likes somebody.
(b) Married people love their spouses.
(c) Everybody is not liked by some person.
(d) People with children like some of them.
(e) No two married people have the same wife.
(f) Not everybody has sons.
(g) There is a man who nobody likes.
(h) Some married people do not like each other.
(i) You must love somebody to like them.
(j) You can’t be your own spouse.
(k) You are your spouse’s spouse.
(l) John likes Mary but she does not like him.
(m) Mary likes whomever John likes.
(n) John is liked by a single person.
(o) John has sons.
(p) John has only sons.

6. Add assertions to verify if the following statements hold in your model (that is, are a consequence of the model’s constraints). Again, if you find one of the statements ambiguous clarify in an Alloy comment how you are disambiguating it.

For each statement, figure out on your own if it holds or not and then use the Alloy analyzer to verify your answer. Provide a short explanation of why the assertion holds or not by referring to the model (signatures and facts).

If you think that an assertion does not hold but the analyzer fails to find a counterexample, consider increasing the scope of the corresponding `check` statement; then report as well the analyzer’s answer in a comment.
(a) Everybody has a mother.
(b) John does not like himself.
(c) Mary does not like herself.
(d) Wives are women.
(e) A woman’s spouse is a man.
(f) You can have at most two (biological) parents.

Warning. In the last two problems above pay particular attention to what quantifiers, if any, are actually meant by each sentence.

B Airport Model

We will model an airport in Alloy 4. For the purposes of this assignment, we can consider an airport to have two basic kinds of resources: fixed resources and mobile resources. Fixed resources include gates, runways, and taxi-ways, which we can collectively refer to as locations. Mobile resources include planes and service vehicles (trucks for delivering fuel/food/etc.) which we can classify as equipment. A basic Alloy model for this domain is the following:

```
abstract sig Resource {}
abstract sig Location extends Resource {}
abstract sig Equipment extends Resource {
    loc: Location
}
sig Gate, Runway, Taxiway extends Location {}
sig Vehicle extends Equipment {}
sig Plane extends Equipment {
    status: Status,
    assignedGate: Gate
}
abstract sig Status {}
one sig OnGround, Flying extends Status {}
```

The model represents the state of an airport at a particular time. The relation \texttt{loc} associates equipment to its location in the airport. The relation \texttt{status} specifies for each plane whether it is flying or on the ground. The relation \texttt{assignedGate} associates planes to the gate they are supposed to go to or be at when on the ground.

1. Modify this basic model as needed to encode the following physical environment constraints and policy constraints on resources:

(a) Any piece of equipment is at most at one place at a time.
(b) Only planes can be on runways.
(c) Gates can have no more than a plane at a time.
(d) Gates can have multiple service vehicles at the same time.
(e) When a plane is at a gate, at least a vehicle is there to service it.
(f) A plane on the ground always has an assigned gate.
(g) A plane cannot be at a gate other than its assigned gate.
(h) Every plane is either flying or on the ground.
(i) Flying planes cannot be in a location.

For each constraint, clearly indicate using Alloy comments which of your addition(s) is encoding it.

2. Add an assertion for each of following statements. Use the AA to verify if the statement is a consequence of the model you have written so far or not and report AA’s response in a comment.

You must also provide a short explanation of why the assertion holds or not by referring to the model (signatures and facts).

(a) No service vehicle can be at a gate with no planes.
(b) Flying planes do not have an assigned gate.
(c) There can be at most one plane on a runway.
(d) Every vehicle has a location.
(e) Planes with different status can be assigned to the same gate.
(f) If a service vehicle is at a gate then a plane is at that gate.

3. Define a parameterless predicate Scenario1 encoding the following scenario for the airport.

- Plane AA36 and luggage truck L5 are at gate G3. Plane AA36 has its assigned gate as G3.
- Fuel vehicle F1 and luggage truck L3 are on taxiway T7.
- Plane AA72 is flying.
- Plane UAL986 is on runway R3 and its assigned gate is G5.

4. For each of the following assertions in hw1b.als, write before it in an Alloy comment a sentence that describes in plain English the constraint after the implication symbol. Also specify in the comment whether the assertion holds or not in the model, considering the facts from question 1, for a scope of 15, by writing (Valid) or (Invalid).

```alloy
assert a1 { Scenario1 => all e: Equipment | one e.loc }
assert a2 { Scenario1 => all l: Location | one loc.l }
assert a3 { Scenario1 => some (AA36 + UAL986) & loc.Gate }
assert a4 { Scenario1 => no l : Vehicle | l in AA36.loc.~loc }
```

1That is, understandable by the average person on the street.
assert a5 { Scenario1 =>
    no p1: Plane, p2: Plane | (p1 + p2).loc = R3 and p1 != p2 }

assert a6 { Scenario1 => lone assignedGate.Gate & Plane }

assert a7 { Scenario1 =>
    some t: Taxiway | some loc.t & Vehicle and no loc.t & Plane }