8.24 Represent the following sentences in first-order logic, using a consistent vocabulary (which you must define):

   a. Some students took French in spring 2001.

   b. Every student who takes French passes it.

   c. Only one student took Greek in spring 2001.

   d. The best score in Greek is always higher than the best score in French.

   e. Every person who buys a policy is smart.

   f. No person buys an expensive policy.

   g. There is an agent who sells policies only to people who are not insured.

   h. There is a barber who shaves all men in town who do not shave themselves.

   i. A person born in the UK, each of whose parents is a UK citizen or a UK resident, is a UK citizen by birth.

   j. A person born outside the UK, one of whose parents is a UK citizen by birth, is a UK citizen by descent.

   k. Politicians can fool some of the people all of the time, and they can fool all of the people some of the time, but they can't fool all of the people all of the time.

   l. All Greeks speak the same language. (Use $Speaks(x, l)$ to mean that person x speaks language l.)

Solution:

Let the basic vocabulary be as follows:

$Takes(x, c, s)$: student x takes course c in semester s;

$Passes(x, c, s)$: student x passes course c in semester s;

$Score(x, c, s)$: the score obtained by student x in course c in semester s;

$x > y$: x is greater than y;

$French$ and $Greek$: specific French and Greek courses;

$Buys(x, y, z)$: x buys y from z (using a binary predicate with unspecified seller is OK but less felicitous);

$Sells(x, y, z)$: x sells y to z;

$Shaves(x, y)$: person x shaves person y;

$Born(x, c)$: person x is born in country c;

$Parent(x, y)$: x is a parent of y;

$Citizen(x, c, r)$: x is a citizen of country c for reason r;

$Resident(x, c)$: x is a resident of country c;

$Fools(x, y, t)$: person x fools person y at time t;

$Student(x)$, $Person(x)$, $Man(x)$, $Barber(x)$, $Expensive(x)$, $Agent(x)$, $Insured(x)$, $Smart(x)$, $Politician(x)$, $IsGreek(x)$: predicates satisfied by members of the corresponding categories.

a. $\exists s \ Student(s) \land Takes(s, French, Spring2001)$

b. $\forall s, t \ Student(s) \land Takes(s, French, t) \Rightarrow Passes(s, French, t)$

c. $\exists s_1 \ Stundent(s_1) \land Takes(s_1, Greek, Spring2001) \land [\forall s_2 \ Student(s_2) \land Takes(s_2, Greek, Spring2001)$
$\Rightarrow s_2 = s_1]$

d. $\forall s \ \exists x \ \forall y \ Score(x, Greek, s) > Score(y, French, s)$

e. $\forall x \ Person(x) \land (\exists y, z \ Policy(y) \land Buys(x, y, z)) \Rightarrow Smart(x)$

f. $\neg(\exists x, y, z \ Person(x) \land Policy(y) \land Expensive(y) \land Buys(x, y, z))$

g. $\exists x \ Agent(x) \land [\forall y, z \ Policy(y) \land Sells(x, y, z) \Rightarrow (Person(z) \land \neg Insured(z))]$

h. $\exists x \ Barber(x) \land [\forall y \ Man(y) \land \neg Shaves(y, y) \Rightarrow Shaves(x, y)]$

i. $\forall x \ Person(x) \land Born(x, UK) \land (\forall y \ Parent(y, x) \Rightarrow (\exists r \ Citizen(y, UK, r) \lor Resident(y, UK))) \Rightarrow$
$Citizen(x, UK, Birth)$

j. $\forall x \ Person(x) \land \neg Born(x, UK) \land (\exists y \ Parent(y, x) \land Citizen(y, UK, Birth)) \Rightarrow Citizen(x, UK, Descent)$

k. $\forall x \ Politician(x) \Rightarrow ((\exists y \ \forall z \ Person(y) \land Fools(x, y, z)) \land (\exists z \ \forall y \ Person(y) \Rightarrow Fools(x, y, z)) \land$
$\neg(\forall y, z \ Person(y) \Rightarrow Fools(x, y, z)))$

l. $\forall x, y, l \ IsGreek(x) \land IsGreek(y) \land Speaks(x, l) \Rightarrow Speaks(y, l)$

9.4 For each pair of atomic sentences, give the most general unifier if it exists:

a. $P(A, B, B), P(x, y, z).$

b. $Q(y, G(A, B)), Q(G(x, x), y).$

c. $Older(Father(y), y), Older(Father(x), John).$

d. $Knows(Father(y), y), Knows(x, x).$

a.
P(A, B, B), P(x, y, z) : {x/A},
P(A, B, B), P(A, y, z) : {x/A, y/B},

P(A, B, B), P(A, B, z) : {x/A, y/B, z/B}

b. Fail

Q(y, G(A, B)), Q(G(x, x), y) : {y/G(x, x)},

Q(G(x, x), G(A, B)), Q(G(x, x), G(x, x)) : {y/G(x, x), x/A},

Q(G(A, A), G(A, B)), Q(G(A, A), G(A, A)) : {y/G(x, x), x/A} Cannot unify any further

c.

Older(Father(y), y), Older(Father(x), John) : {x/y}

Older(Father(x), x), Older(Father(x), John) : {y/x, x/John} = {y/John, x/John}

d.

Knows(Father(y), y), Knows(x, x) : {x/Father(y)}

Knows(Father(y), y), Knows(Father(y), Father(y)) : {x/Father(y)} Cannot unify any further