p647
WFC
p1398

Prolog Question 2000

Answer all parts.

One of the regulations of the International Rugby Board (IRB) states that for a player to be eligible to play for a given country, the player's father or mother or grandfather or grandmother must have been born in that country. Assume that there is a complete genealogical database consisting of Prolog clauses of the form **person(P, B, F, M)**, where **P** is a person's name, **B** is the country of **P**'s birth, **F** is their father's name and **M** is their mother's name. For example, the clause

person(bruce, australia, rhodri, bronwyn).

might appear in such a database. Further assume that names in the database are constructed so as to refer uniquely to individuals. Write Prolog clauses defining the predicate **eligible** such that goals of the form **eligible(P,C)** succeed if and only if person **P** is eligible to play for country **C** according to the above regulation. [10 marks]

Given a list of players on a given country's team, define a predicate **checkteam** that will check each member of the team for eligibility according to the **eligible** predicate, and furthermore check that each player appears on the list only once. The **checkteam** goal will fail if any player is ineligible or if any player is listed more than once. [10 marks]

Answer

Note that the regulation does not require the player to be born in the country. Here is the most straightforward definition:

```
eligible(P,C) :- person(P,_,F,_), person(F,C,_,_).
eligible(P,C) :- person(P,_,_,M), person(M,C,_,_).
eligible(P,C) :- grandparent(P,C).

grandparent(P,C) :- person(P,_,F,_), person(F,_,GF,_), person(GF,C,_,_).
grandparent(P,C) :- person(P,_,F,_), person(F,_,_,GM), person(GM,C,_,_).
grandparent(P,C) :- person(P,_,_,M), person(M,_,GF,_), person(GF,C,_,_).
grandparent(P,C) :- person(P,_,_,M), person(M,_,_,GM), person(GM,C,_,_).
```

Note checkteam should succeed only for non-null teams.

```
checkteam([H|T], C) :- ct([H|T], C), allunique([H|T]).

ct([], C) :- !.
ct([H|T], C) :- eligible(H, C), ct(T, C).

allunique([]).
allunique([H|T]) :- notmember(H, T), !, allunique(T).

notmember(_,[]) :- !.
notmember(X, [H|T]) :- X \== H, notmember(X, T).
```