Z Specification – Exercises

[Questions from the 2006 exam, Thanks to Fisher]

1. This question concerns the basic structures used within Z specifications.

   (a) In Z, what is the difference between a *total* function and a *surjection*?

   (b) Given a bag

   $$B == [\![ford, toyota, fiat, ford, toyota, honda, ford]\!]$$

   then what is $\mathrm{dom}\, B$ and what is $\mathrm{ran}\, B$? How will $\mathrm{dom}\, B$ and $\mathrm{ran}\, B$ change if we add another '$fiat$' to B?

   (c) If f and g are both functions of the same type, then under what circumstances (if any) will

   $$\mathrm{dom}(f \cup g) = (\mathrm{dom}\, f) \cup (\mathrm{dom}\, g) \ ?$$

   Give an appropriate example to illustrate your answer.

   (d) Given $r : \mathbb{P}\{apple, banana, grape, peach\}$, write down all the values r can possibly have.

2. We are developing a Z specification for a family tree and have developed the initial state space schema below (where $PEOPLE$ is the set of all people):

   Ancestry

   $is\_parent\_of : PEOPLE \pfun \mathbb{P}\, PEOPLE$

   ...

   Here, $is\_parent\_of(fred) = \{emily, david\}$ if $(fred \mapsto \{emily, david\}) \in is\_parent\_of$.

   (a) What invariant might we add to the above state space schema in order to ensure that no one is a parent of themselves?

   (b) Write a Z specification for an *AddChild* operation, which adds a child for an already existing parent.

   The operation should take inputs $child? : PEOPLE$ and $parent? : PEOPLE$.

   (c) How would you describe, using Z notation, the set of all people who are *grandparents*?

   (d) Write a Z specification for a *GetChildren* operation, which returns the set of children for a given parent.

   The operation should take $parent? : PEOPLE$ as input and return $children! : \mathbb{P}\, PEOPLE$ as output.