2008 Paper 10 Question 6

Introduction to Functional Programming

(*a*) Write an SML function

```
propercuts:  (α list) → (α list * α list) list
```

that given a list ℓ outputs the list of all pairs of non-empty lists (ℓ_1, ℓ_2) such that ℓ_1@$\ell_2 = \ell$. [5 marks]

(*b*) Consider the following datatypes

```
datatype α tree = leaf of α | node of α tree * α tree ;

datatype α symbol = Lbracket | Rbracket | token of α ;
```

and the SML function

```
rep:  α tree → (α symbol) list
```

that represents a binary tree as a list of symbols, according to the following definition:

```
fun rep ( leaf x ) = [ token x ]
  | rep ( node(l,r) )
    = [ Lbracket ] @ (rep l) @ (rep r) @ [ Rbracket ] ;
```

Write an SML function

```
istree:  (α symbol) list → bool
```

that given a list of symbols ℓ outputs `true` if there exists a (necessarily unique) tree t such that `rep`$(t) = \ell$, and outputs `false` otherwise. [10 marks]

(*c*) Define the SML functions

```
infix @ ;
@: α list * α list → α list ;

map: (α → β) → α list → β list ;
```

and rigorously argue for the correctness of the following identity:

```
map f (ℓ1 @ ℓ2) = (map f ℓ1) @ (map f ℓ2)   : β list
```

for all $f : \alpha \to \beta$ and $\ell_1, \ell_2 : \alpha$ `list`.

[5 marks]