

Homework 3 solutions

Name: Hung Nguyen

B-ID: B01037287

Question 1

1a `[0|1]`.

This Prolog expression is invalid because after `|` should be a list.

1b `[0, 1]`.

```
0:1: []
```

This is allowed by Haskell.

1c `[0|[1]]`.

```
0:1: []
```

This is allowed by Haskell.

1d `[0, [1]]`.

This is not allowed by Haskell because lists must be homogeneous.

1e `[0|[1|[2|[]]]]`.

```
0:1:2[]
```

This is allowed by Haskell.

Question 2

2a `p1([_|X], X)`.

```
p1 x = tail x
```

2b `p2([_, _|X], X)`.

```
p2 x = tail (tail x)
```

2c `p3([_, X|_], X).`

```
p3 x = head (tail x)
```

2d `p4([_, [X|_] | _], X).`

```
p4 x = head (head (tail x))
```

2e `p5([_, [_, X] | _], X).`

```
p5 x = head (tail (head (tail x)))
```

Question 3

- Prolog uses a declarative evaluation strategy known as backtracking. It explores all possible solutions to a problem until it finds one that satisfies the query.
- Haskell functions, on the other hand, are typically evaluated using a lazy evaluation strategy. Haskell computes values as they are needed, rather than eagerly computing them all at once. This can lead to more efficient execution in some cases but can be challenging to reason about compared to Prolog's backtracking.***

Question 4

Because lists must be homogeneous, and different items in the list might not have the same type.

Question 5

X on the left hand side is an array because of the `|` operator. To get X to be the value of `1`, we need to replace `|` with `,`

```
g([0|X], 1) = g([0, 1], X)
```

Question 6

6a

`root(A, B, C, X): X`

6b

`split(Xs, Ys, Zs): Ys, Zs`

6c

`split(Xs, D, Ys): Ys`

Question 7

7a

A relation between two sets A and B is defined as any subset of the Cartesian product $A \times B$. Since $A = \{1, 2, 3\}$ and $B = \{a, b\}$, the Cartesian product $A \times B$ consists of $3 \times 2 = 6$ pairs.

7b

A function from A to B is a relation where each element of A maps to exactly one element of B. Since A has 3 elements and B has 2 elements, there are $2^3 = 8$ possible functions from A to B.

7c

Question 8

8a: By value

```
15
15
```

8b: By reference

```
25
25
```

8c: Copy-Restore

```
15
3
```

Question 9

```
a (b (c d e))
```

- $c\ d\ e$ applies the function c to arguments d and e .
- $b\ (c\ d\ e)$ applies the function b to the result of $c\ d\ e$.
- $a\ (b\ (c\ d\ e))$ applies the function a to the result of $b\ (c\ d\ e)$.

Question 10

10a

Invalid. Languages like C, which typically push arguments onto the stack left-to-right, still support variable argument functions using `va_start` and `va_arg`.

10b

Not entirely true, humans are more used to going over a list of items from left to right rather than right to left. However, depends on the problem, it might be useful to make use of right-associativity of `foldr`.

10c

Valid. Prolog list is more flexible because Haskell list is homogeneous.

10d

Not every relation is necessarily a function. For example, a relation that maps names to ages in a database might have multiple people with the same name but different ages, violating the condition for a function, which maps same name to 1 age.

10e

Depends on the programming languages. In C++, if a function returns a reference, then the result of the function call can be an L-value.