

1. (8 p)

Lösning:

1. Mål: `conc(X, Y, [a])`.
2. Skapar instans av första regeln:
`conc([], X1, X1)`.
Unifierar: $X = [], X1 = [a], Y = [a]$.
Första svaret: $X = [], Y = [a]$.
Backtrackar till 2.
3. Skapar instans av andra regeln:
`conc([H2 | T2], X2, [H2 | TX2]) :- conc(T2, X2, TX2)`.
Unifierar $X = [a | T2], X2 = Y, H2 = a, TX2 = []$.
 - 3.1. Delmål: `conc(T2, Y, [])`.
 - 3.2. Skapar instans av första regeln:
`conc([], X3, X3)`.
Unifierar: $T2 = [], X3 = [], Y = []$.
Andra svaret: $X = [a], Y = []$.
Backtrackar till 3.2.
 - 3.3. Skapar instans av andra regeln:
`conc([H4 | T4], X4, [H4 | TX4]) :- conc(T4, X4, TX4)`.
Kan inte unifiera `[]` med `[H4 | TX4]`.
Backtrackar till 3.
4. Hittar inga andra regler för `conc`.
Tredje och sista svaret: `false`.

Rättning:

- 1p för rätt huvuddrag, dvs backtracking på rätta ställen och rätt antal svar.
- 1p för rätt regelinstans och unifiering vid steg 2.
- 1p för rätt första svar vid steg 2.
- 1p för rätt regelinstans och unifiering vid steg 3 (innan 3.1.).
- 1p för rätt regelinstans och unifiering vid steg 3.2.
- 1p för rätt andra svar vid steg 3.2.
- 1p för rätt regelinstans och (ingen!) unifiering vid steg 3.3.
- 1p för rätt tredje och sista svar vid steg 4.

2. (12 p)

Lösning:

- (a)

```
assignRandom([], _, []).
assignRandom([HF | TF], To, [(HF, X) | TM]) :-
    member(X, To),
    assignRandom(TF, To, TM).
```
- (b)

```
checkAssignment([]).
checkAssignment([(_, Y) | T]) :-
    \+ member( (_, Y), T),
    checkAssignment(T).
```
- (c)

```
assignInjective([], _, []).
assignInjective([HF | TF], To, [(HF, X) | TM]) :-
    select(X, To, ToRest),
    assignInjective(TF, ToRest, TM).
```

Rättning:

- (a) 1p för rätt basfall (tom student-lista),
1p för rätta mönster i huvudet av andra regeln,
1p för rätt applicering av `member`, och
1p för rätt rekursivt anrop.
- (b) 1p för rätt basfall (tom tilldelning),
1p för rätt mönster i huvudet av andra regeln,
1p för rätt applicering av `member` (negerat!), och
1p för rätt rekursivt anrop.
- (c) 1p för rätt basfall (tom student-lista),
1p för rätta mönster i huvudet av andra regeln,
1p för rätt applicering av `select`, och
1p för rätt rekursivt anrop.