Unification in Prolog

For each of the unification attempts below, state:
- Whether or not the unification succeeds
- If the unification succeeds, state the values of each variable

1.) `1 = 1`

`true`

2.) `1 = 2`

`true`

3.) `X = 27`

`X = 21`

4.) `1 = X`

`X = 1`

5.) `X = foo`

`X = foo`

6.) `foo = bar`

`false`

7.) `1 = baz`

`false`

8.) `foo(1) = foo(1)`

`true`

9.) `foo(1) = foo(2)`

`false`

10.) `foo(X) = foo(1)`

`X = 1`

11.) `foo(1) = foo(X)`

`X = 1`

12.) `foo(1) = foo(1, 2)`

`false`

13.) `foo(X, Y) = foo(1)`

`false`

14.) `foo(X, Y) = foo(1, 2)`

`X = 1, Y = 2`

15.) `foo(1, Y) = foo(X, 2)`

`X = 1, Y = 2`

16.) `foo(1, 2) = foo(X, X)`

`false`

17.) `foo(bar(X), Y) = foo(Z, bar)`

`Y = bar, Z = bar(X)`

18.) `foo(bar(X), foo(Y)) = foo(foo(1), foo(2))`

`false`

19.) `foo(bar(X), foo(2)) = foo(bar(3), foo(Y))`

`X = 3, Y = 2`

20.) `foo(bar(X), X) = foo(Y, 2)`

`X = 2, Y = bar(2)`

21.) `foo(1, foo(2, foo(3, bar))) = foo(1, foo(2, foo(bar)))` `false`