1.
For example, let's say we have the following SAT instance:
V = {x, y, z}
A = (x ∨ ¬y) ∧ (¬x ∨ y ∨ z)
We can reduce this to a CSP instance P' as follows:
V' = {x, y, z}
D' = {0, 1}
C' = {(x, {0, 1}), (y, {0, 1}), (z, {0, 1}), (x ∨ ¬y, {0}), (¬x ∨ y ∨ z, {0})}

We can then define the bijection f as follows:
f: {x=1, y=0, z=1} -> {x=1, y=0, z=1}
This bijection maps a satisfying assignment of the SAT instance (i.e., {x=1, y=0, z=1}) to a solution of the CSP instance (i.e., an assignment of truth values to the variables x, y, and z in the CSP instance).

The inverse bijection f' would then be defined as follows:
f': {x=1, y=0, z=1} -> {x=1, y=0, z=1}
This bijection maps a solution of the CSP instance (i.e., an assignment of truth values to the variables x, y, and z) back to a satisfying assignment of the SAT instance (i.e., {x=1, y=0, z=1}).

2.
consider the following CSP instance with three variables a, b, and c, each with a domain of {1, 2, 3}. The constraints are:

- $a \neq b$
- $a \neq c$
- $b \neq c$

To define a SAT instance (V', A') from this CSP instance, we can follow these steps:

1. Create Boolean variables a', b', and c' in V'.
2. Create Boolean variables $d1'$, $d2'$, and $d3'$ in V'.
3. Create the following conjunctions in A':

- $(\neg a' \vee \neg b')$
- $(\neg a' \vee \neg c')$
- $(\neg b' \vee \neg c')$

This results in the following SAT instance (V', A'):

- $V' = \{a', b', c', d1', d2', d3'\}$
- $A' = \{(\neg a' \vee \neg b'), (\neg a' \vee \neg c'), (\neg b' \vee \neg c')\}$

Now, let's say we have a satisfying assignment for this SAT instance that sets the variables as follows:

- a' = True
- b' = False
- c' = True
- $d1'$ = True
- $d2'$ = False
- $d3'$ = False

The function f maps this satisfying assignment to a solution for the original CSP instance as follows:

- a = 1 (because $d1'$ is True)
- b = 2 (because $d2'$ is False and b' is False)
- c = 3 (because $d3'$ is False and c' is True)

the inverse function f' would map the solution a = 1, b = 2, c = 3 for the CSP instance to the following satisfying assignment for the SAT instance:

- a' = True (because 1 is the value of a in the solution and $d1'$ is True)
- b' = False (because 2 is the value of b in the solution and $d2'$ is False)
- c' = True (because 3 is the value of c in the solution and $d3'$ is False)
- $d1'$ = True (because 1 is the value assigned to a in the solution)
- $d2'$ = False (because 2 is the value assigned to b in the solution)
- $d3'$ = False (because 3 is the value assigned to c in the solution)

This inverse function f' can be used to determine whether a given solution for the CSP instance is a valid satisfying assignment for the corresponding SAT instance.