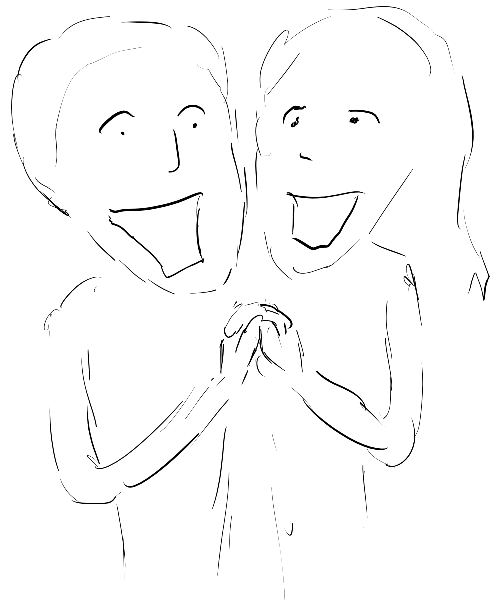
$P$: set of people

$N$: set of social needs

$S(p_1, n, p_2)$: $p_1$ meets need $n$ for $p_2$

Soul Mate Hypothesis

$$\boxed{\forall x \in P.\ \exists y \in P.\ \forall n \in N.\ x \neq y \wedge S(x,n,y) \wedge S(y,n,x)}$$



domain: People at UVA

$F(x)$: $x$ is faculty

$E(x)$: $x$ is an employee

1 $\forall x.\ \boxed{F(x)} \to E(x)$

$B$: set of UVA Faculty

$E(x)$: $x$ is an employee of UVA

2 $\forall \boxed{x \in B}.\ E(x)$

$C$: set of UVA employees

3 $\forall x \in B.\ x \in C$

4 $\boxed{B \setminus C = \{\}}$

5 $\forall x.\ (x \in B \to x \in C)$

6 $\boxed{B \subseteq C}$

7 $C \supseteq B$

8 $B \cap C = B$

9 $B \cup C = C$

$$\boxed{\begin{array}{l} S = \{x \mid P(x)\} \\ P(x):\quad x \in S \end{array}}$$

$$\boxed{\begin{array}{l} T = \{(x,y,z) \mid Q(x,y,z)\} \\ Q(x,y,z):\quad (x,y,z) \in T \end{array}}$$

$\{x+2 \mid \_\_\_\_\}$

$A \subseteq B \quad \equiv \quad \forall x \in A.\ x \in B$

$A \cup B = \{x \mid x \in A \lor x \in B\}$

$\forall x.\ (x \in A \lor x \in B) \leftrightarrow (x \in A \cup B)$

$M = \{f(x) \mid P(x)\}$

$\forall x.\ P(x) \leftrightarrow (f(x) \in M)$

$O$: set of OS

$F$: set of flows

$\boxed{H(x,y): x \text{ has } y}$

$H(x,y)$

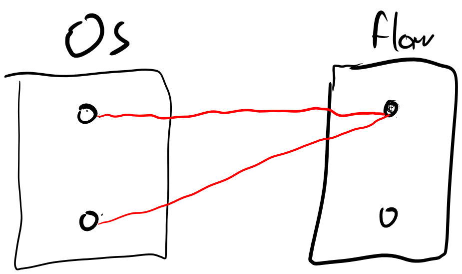
$$\underbrace{\text{Every}}_{\forall x \in O} \text{ OS has } \underbrace{\boxed{\text{a}} \text{ flow}}_{\exists y \in F}$$

$\forall x \in O.\ \exists y \in F.\ H(x,y)$

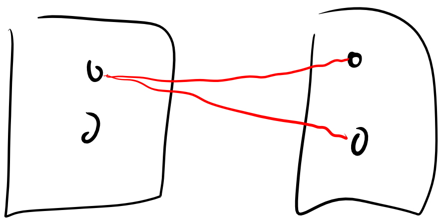
$H(x,y)$

there is a flow that every OS has

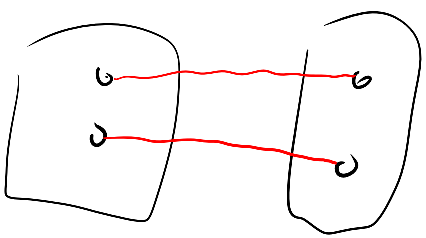
$\exists y \in F.\ \forall x \in O.\ H(x,y)$

1. 

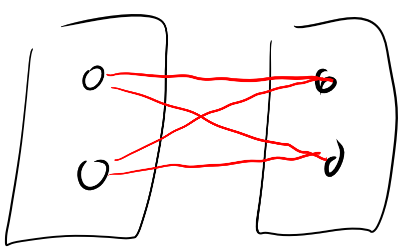

Yes | yes

2. 

NO | no

3. 

Yes | no

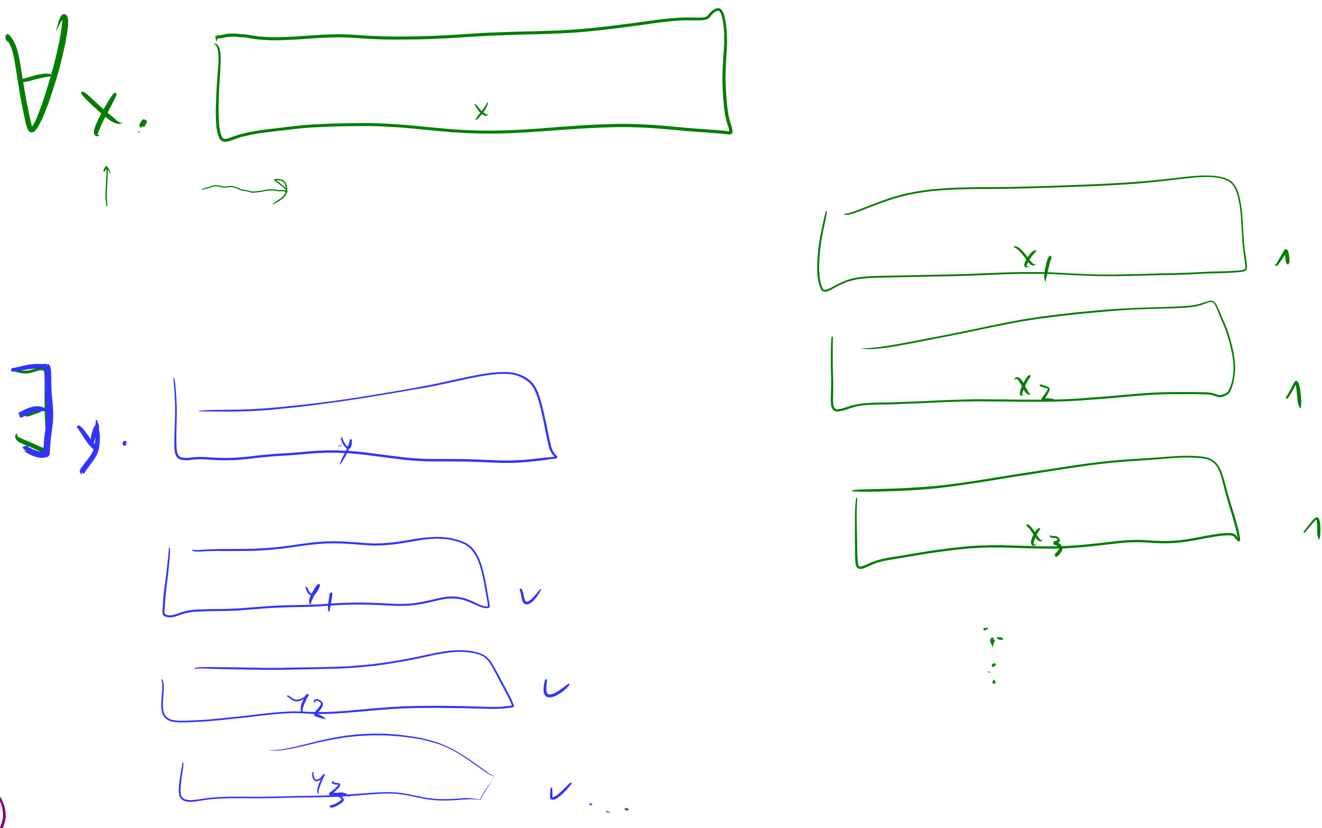
4. 

Yes | yes

x = BeOS    y = no one uses it

$\forall x \in O.\ \exists y \in F.\ H(x,y)$

$\exists y \in F.\ \forall x \in O.\ H(x,y)$



$\forall x \in A . \exists y \in B . P(x,y)$

```
ans = T
for each x in A:
    ans_2 = ⊥
    for each y in B:
        if P(x, y) : ans_2 = T
    if ¬ ans_2 : ans = ⊥
return ans
```

$\forall x \in A . P(x)$

```
ans = T
for each x in A:
    if ¬P(x): ans = ⊥
return ans
```

$\exists x \in A . P(x)$

```
ans = ⊥
for each x in A:
    if P(x): ans = T
return ans
```