[Add Title]

[Add Author name]

June 2021

1 Algorithm

1.1 Implementation for each Atom

The following list mentions how each of the atoms are dealt with to create the Feature tree.

1. $x[f]y$: When ever it is encountered a mapping is added to node of x from f to y.

2. $x[f] \uparrow$: Use the special variable 0 (which represents absent) and execute atom x[f]var0

3. $x =_* y$: union the node of x and that of y and map both the variables to this node. In the node keep record of all the variables it is mapped from

4. $x =_F y$: During the initial loop of the clause keep a record of the set F and the other variable in node of x and that of y. Now during equality dissolve phase.[Note: write about this part in a different section] Go through each node and and for all features it has an equality find its mapping. This mapping can be present in the current node itself, in the node of the variable the equality is with, or an equality this node on current feature and so on. If no such mapping is present we go on to the next equality on the feature. If till the end we get no mapping we skip the current feature and go for the next.

5. $x[F]$: Check all features that have a mapping in x. And if any such feature f $\notin$ F then the input clause is not valid (there is a clash). This check needs to be done after equality dissolve

6. $x \neq_F y$: Go through each feature in set F, if any of them give a different mapping in node of x and node of y. Then the atom is already satisfied. One of the node is having a mapping for a feature and other does not. In that case for the other node add a absent mapping for the feature. If both cases don't satisfy find an element f $\in$ F which has no mapping in node of x or that of y. And create a mapping with a new feature and variable to x and an absent mapping to y. If not such f exists then it is a clash(invalid input clause) [Note to self: Think of cases where this solution wouldn't work]

7. $\neg x[F]$: Check if for any f $\notin$ F there is a mapping in node of x. If so, atom is already satisfied. If not, create a new feature and variable and add a map to node of x for them.

8. $\phi[x \rightarrow y]$: [Note to self: I think its just for representation purpose, should be same as $x =_* y$, check rules]

9. $x \sim_F y$: This atom means nodes of x and y have same mapping everywhere except maybe in some elements(features) of set F. [To be written]

10. $x \nsim_F y$: This atom means x and y differ in s feature f $\notin$ F. Check for a feature $f \notin F$ which has a mapping in node of x what differs in node of y [if feature has no mapping in node of y we can add an absent]. If no such feature is found do the same with roles of x,y interchanged. If either is satisfied then the atom is satisfied by our current tree. If not create a new feature and a new variable and map them in node of x and an absent map in y for the feature. If there exists a x[E] in the clause, the step mentioned above cant be performed , so we replace the roles of x and y in previous step. If y also has a y[G]. We will have to find a feature $f \in [E - F]$ which has no specific mapping in x and add a different mapping than in y(add absent in node of y for f, if no mapping present).If no such f exists try with y. If not luck even then, its a crash(invalid input clause)