

```

1 ncpd goal = residualize o drive o normalize (goal)
2 drive      = drive_disj ∪ drive_conj
3
4 drive_disj :: Disjunction → Process_Tree
5 drive_disj D@(c1, ..., cn) =
6     create_or_node [ci ← drive_conj ci]
7
8 drive_conj :: (Conjunction, Substitution) → Process_Tree
9 drive_conj (C@(r1, ..., rn), subst) =
10     r1, ..., rn ← propagate substitution subst on r1, ..., rn
11     switch whistle (C) of
12         instance (C', subst')          → create_fold_node (C', subst')
13         embedded_but_not_instance → create_stop_node (C, subst)
14     otherwise →
15         r ← select_a_call (r1, ..., rn)
16         r' ← drive o normalize o unfold (r)
17         if trivial o leafs (r')
18         then
19             C' ← propagate_subst (C \ r, extract_subst r')
20             drive C'[r ↦ r']
21         else
22             (r', subst') ∧ drive (C \ r, subst)

```