```
(set-logic ALL)

(set-option :produce-models true)
(set-option :incremental true)
(set-option :finite-model-find true)
(set-option :sets-ext true)

(declare-sort Machine 0)

(declare-fun fc (Machine) Real)
(declare-fun ridctr (Machine) Real)
(declare-const maxh Real)

(define-fun human_made ((m1 Machine)) Bool
    (< (fc m1) maxh)
)

(declare-const all_perceptual_experience Bool)

(assert (forall ((m1 Machine) (m2 Machine))
        (=> all_perceptual_experience
                (=> (and (> (ridctr m1) 0) (> (fc m2) (fc m1))
                    )
                    (> (ridctr m2) (ridctr m1)))
        )
        )
)

(assert (forall ((m1 Machine))
            (=> (human_made m1)
                (> (ridctr m1) 0))
        )
)


(echo "fc = functional_complexity")
(echo "maxh = max human made machines fc value")
(echo "ridctr =
required_intelligence_design_and_tecnichal_creativity_to_reprod
uce")
(echo "ASSERTIONS BEFORE")
(get-assertions)
(check-sat)

```

```
42 (push 1)
43 (assert (not
44         (forall ((m1 Machine) (m2 Machine))
45         (=>
46                 (and    all_perceptual_experience (human_made
                   m1) (> (fc m2) maxh) )
47                 (> (ridctr m2) (ridctr m1))
48         ))
49         )
50 )
51
52 (echo "ASSERTIONS AFTER")
53 (get-assertions)
54 ; (get-assignment)
55
56 (check-sat)
57 ; (get-model)
58
59 (pop 1)
60
61
```