```
                    (backchain goal rules facts)


user=> (backchain 'e '((c a b) (e c d)) '(a b d))
TRACE t254: (backchain e ((c a b) (e c d)) (a b d))
TRACE t255: | (member e (a b d))
TRACE t255: | => nil
            ; trying rule (e c d): e if c and d
TRACE t259: | (backchain c ((c a b) (e c d)) (a b d))
TRACE t260: | | (member c (a b d))
TRACE t260: | | => nil
              ; trying rule (c a b): c if a and b
TRACE t264: | | (backchain a ((c a b) (e c d)) (a b d))
TRACE t265: | | | (member a (a b d))
TRACE t265: | | | => (a b d)
TRACE t264: | | => (a b d)
TRACE t266: | | (backchain b ((c a b) (e c d)) (a b d))
TRACE t267: | | | (member b (a b d))
TRACE t267: | | | => (b d)
TRACE t266: | | => (b d)
TRACE t259: | => true
TRACE t269: | (backchain d ((c a b) (e c d)) (a b d))
TRACE t270: | | (member d (a b d))
TRACE t270: | | => (d)
TRACE t269: | => (d)
TRACE t254: => true
true
```

Fact = Rule with No Premises

```
(def clauses '((a) (b) (d) (c a b) (e c d)) )

user=> (backch 'e)
TRACE t290: (backch e)
TRACE t291: | (backch c)
TRACE t292: | | (backch a)
TRACE t292: | | => true
TRACE t293: | | (backch b)
TRACE t293: | | => true
TRACE t291: | => true
TRACE t294: | (backch d)
TRACE t294: | => true
TRACE t290: => true
true
```