Negation as Failure

Consider the example in which we define "even" as the opposite of being "odd":

Logically, we would like to have the rule

even(X) :- ¬ odd(X).

In predicate logic, this is $\forall \mathbf{X}(\neg\ \mathbf{odd}(\mathbf{X}) \quad \rightarrow \quad \mathbf{even}(\mathbf{X}))$

In real Prolog we have even(X) :- not odd(X).

To solve "not odd(X)", Prolog looks for *any* solution for "odd(X)"; if there exists a solution, "not odd(X)" fails.

Thus, the Prolog rule really means in logic:

$\neg\ \exists \mathbf{X}\ \mathbf{odd}(\mathbf{X}) \quad \rightarrow \quad \forall \mathbf{X}\ \mathbf{even}(\mathbf{X}))$

which is $\forall \mathbf{X}\ \neg\ \mathbf{odd}(\mathbf{X}) \quad \rightarrow \quad \forall \mathbf{X}\ \mathbf{even}(\mathbf{X}))$

(∀ doesn't distribute over ∨, thus not over →)

It's nonsense. But it is true for a particular constant value of X.

$$\neg\ \mathbf{odd}(\mathbf{c}) \quad \rightarrow \quad \mathbf{even}(\mathbf{c}))$$