I illustrate the reductions from PCP to unsatisfiability and to existence of a finite model using an example instance of PCP. It should be not difficult to derive the general construction from that.

Consider the following PCP-instance

\[ V = (a, bc, ba), \quad W = (ab, bc, a). \]

It has a single solution \((1, 3)\), which results in the word \((aba)\).

**Reduction to Unsatisfiability**

If you know Prolog, or resolution, the construction is very easy to understand.

We use the following symbols:

- A constant \(\lambda\), denoting the empty word.
- For each letter \(x\) in the alphabet, a unary function symbol \(x\).
- A constant 0, and a unary function \(s\).
- A ternary predicate symbol \(p\).

Then we translate the problem into:

\[
p(\lambda, \lambda, 0),
\]

\[
\forall xyn p(x, y) \rightarrow p(a(x), b(a(y)), s(n)),
\]

\[
\forall xyn p(x, y) \rightarrow p(c(b(x)), c(b(y)), s(n)),
\]

\[
\forall xyn p(x, y) \rightarrow p(a(b(x)), a(y), s(n)),
\]

\[
\forall xn \neg p(x, x, s(n)).
\]

First prove \(p(\lambda, \lambda, 0)\), then \(p(\lambda, a(\lambda), s(0))\), then \(p(a(b(\lambda))), a(b(\lambda))), s(s(0))\), and after that \(\bot\).

**Reduction to Existence of Finite Model**

We use the following symbols:

- A unary predicate \(\lambda\), denoting the empty word.
- For each letter \(x\) in the alphabet, a binary function symbol \(x\).
In this case, there is a finite model, which can be described as follows:

• A binary predicate symbol $p$.

• We first set a starting point:
  
  $$\exists x \ p(x, x) \land \lambda(x).$$

• Then comes the main translation:
  
  Let $\Phi(x, y) := p(x, y) \land (x \neq y \lor \lambda(x))$.
  Let $C_1(x, y) := \exists x_1 y_1 y_2 a(x, x_1) \land a(y, y_1) \land b(y_1, y_2) \land p(y_1, y_2)$
  Let $C_2(x, y) := \exists x_1 x_2 y_1 y_2 b(x, x_1) \land c(x_1, x_2) \land b(y_1, y_2) \land c(y_1, y_2) \land p(y_1, y_2)$.
  Let $C_3(x, y) := \exists x_1 x_2 y_1 b(x, x_1) \land a(x_1, x_2) \land a(y, y_1) \land p(x_2, y_1)$.

  The main translation is:
  
  $$\forall xy \ \Phi(x, y) \rightarrow C_1(x, y) \lor C_2(x, y) \lor C_3(x, y)$$

  The idea of the construction is that, when $(c, d) \in [p]$, and $c \neq p$, we are forced to select one of the $C_1, C_2, C_3$ which will create two new points $(c', d') \in [p]$, s.t. $c'$ is reachable through a chain from $c$, and $d'$ is reachable through a chain from $d$. The chains correspond to a pair $(v_i, w_i)$ from the original PCP problem. The only way to stop extending forever, is to reach a point $(c'', d'') \in [p]$ with $c'' = d''$.

• And we ensure that every point that is reachable from a point $x$ on which $\lambda(x)$ holds, is reachable in only one way:
  
  (Otherwise, we could for example start in a point $(x, y)$ with $x = y$, make the chain $b(x, x_1), \ a(x_1, x_2)$ end in the same point as the chain $a(x, y_1)$, have $p(x_2, y_1)$, and stop by cheating)

There is no way back to $\lambda$:

$$\forall x_1 x_2 \ a(x_1, x_2) \land \lambda(x_2) \rightarrow \bot,$$
$$\forall x_1 x_2 \ b(x_1, x_2) \land \lambda(x_2) \rightarrow \bot,$$
$$\forall x_1 x_2 \ c(x_1, x_2) \land \lambda(x_2) \rightarrow \bot.$$

Through different predicates $a, b, c$ one cannot reach the same point:

$$\forall xyz \ a(x, z) \land b(y, z) \rightarrow \bot,$$
$$\forall xyz \ a(x, z) \land c(y, z) \rightarrow \bot,$$
$$\forall xyz \ b(x, z) \land c(y, z) \rightarrow \bot.$$

Through each of $a, b, c$ one cannot arrive at the same point from different start positions:

$$\forall x_1 x_2 y_1 y_2 \ a(x_1, x_2) \land a(y_1, y_2) \land x_2 = y_2 \rightarrow x_1 = y_1,$$
$$\forall x_1 x_2 y_1 y_2 \ b(x_1, x_2) \land b(y_1, y_2) \land x_2 = y_2 \rightarrow x_1 = y_1,$$
$$\forall x_1 x_2 y_1 y_2 \ c(x_1, x_2) \land c(y_1, y_2) \land x_2 = y_2 \rightarrow x_1 = y_1.$$

In this case, there is a finite model, which can be described as follows:

$$D = \{d_0, d_1, d_2, d_3\},$$
\[ \lambda = \{d_0\}, \]
\[ A = \{(d_0, d_1), (d_2, d_3)\}, \]
\[ B = \{(d_1, d_2)\}, \]
\[ C = \{\}, \]
\[ p = \{(d_0, d_0), (d_1, d_2), (d_3, d_3)\}. \]