Stratified Programs
Observation:
Every normal program is consistent (has a model), but this is not necessarily true for \( \text{comp}(P) \).

Example:
program \( P \):
\[
\begin{align*}
p & \leftarrow \sim q \\
q & \leftarrow \sim r \\
r & \leftarrow \sim p
\end{align*}
\]
\( I = \{p, q, r\} \) is a model

\( \text{comp}(P) \):
\[
\begin{align*}
p & \leftrightarrow \sim q \\
q & \leftrightarrow \sim r \\
r & \leftrightarrow \sim p
\end{align*}
\]
By transitivity:
\[
\begin{align*}
p & \leftrightarrow \sim p \\
q & \leftrightarrow \sim p \\
r & \leftrightarrow \sim p
\end{align*}
\]
Thus there exists no model for \( \text{comp}(P) \)
Definition:

A *level mapping* of a normal program is a mapping from its set of predicate symbols to the non-negative integers. We refer to the value of a predicate symbol under this mapping as the *level* of that predicate symbol.
Level mapping:
  mapping from a set of relation symbols to \( \mathbb{N} \).
\( l(r) \) is called the level of \( r \).

**Theorem.** Let \( C \) be a finite non-recursive set of clauses.
Then there exists a level mapping \( l \) such that for every clause \( c \in C \),
  if \( q \) occurs in the body of \( c \) and \( c \) defines \( r \),
  then \( l(r) > l(q) \).
Hierarchical Normal Program

**Definition:**
A normal program is hierarchical if it has a mapping such that in every program clause
\[ A \leftarrow L_1, \ldots, L_n, \] the level of every predicate symbol occurring in the body is less than the level of \( A \).

**Observation:**
not hierarchical:
\[ \text{relatedTo}(x,y) \leftarrow \text{relatedTo}(y,x) \]
• **Definition:**

A normal program is stratified if it has a level mapping such that in every clause $A \leftarrow L_1, \ldots, L_n$,

- the level of the predicate symbol of every positive literal is less or equal to the level of $A$ and
- the level of each predicate symbol of every negative literal is less than the level of $A$. 
Example for Stratification

Level 2:
- `loves(x,y) ← friend(x,y)`
- `loves(x,y) ← enemy(x,y)`

Level 1:
- `enemy(x,y) ← ~friend(x,y)`
- `enemy(x,y) ← friend(x,z), enemy(z,y), ~friend(x,y)`

Level 0:
- `friend(x,z) ← friend(x,y), friend(y,z)`
- `friend(a,b) ←`
Counterexample

```
man(x) ← person(x), ~woman(x)
woman(x) ← person(x), ~man(x).
woman(x) ← person(x), ~man(x).
man(x) ← person(x), ~woman(x)
```

Diagram:

```
1 0 2
man(x) ← person(x), ~woman(x)
2 0 1
woman(x) ← person(x), ~man(x).
2 0 1
man(x) ← person(x), ~woman(x)
```
Corollary:
Let $P$ be a stratified normal program. Then $\text{comp}(P)$ has a minimal normal Herbrand model. A normal Herbrand model assigns the equality relation to "=".
**Corollary:**
Let $P$ be a stratified normal program. Then $\text{comp}(P)$ has a minimal normal Herbrand model. A normal Herbrand model assigns the equality relation to „=“. 
**Corollary:**
Let \( P \) be a stratified normal program. Then \( \text{comp}(P) \) has a minimal normal Herbrand model. A normal Herbrand model assigns the equality relation to "=".
Computational counterpart to models of stratified programmes:

Computing with finite failure
Definition
A normal program is locally stratified if each atom in $B_P$ can be assigned a countable ordinal level such that no atom positively depends on an atom of greater level negatively depends on an atom of equal or greater level.
even(s(X)) ← ¬even(X).
even(0).

\[ B_P: \{ \text{even}(0)^0, \text{even}(s(0))^1, \text{even}(s(s(0)))^2, \text{even}(\ldots)^3, \ldots \} \]
even(s(X)) ← ¬even(X).
even(0).
even(0) ← q(X).

\[ B_P: \]
\[ J = \{ q(0)^0, even(0)^1, even(s(s(0)))^3, \ldots \} \]
\[ I = \{ even(0)^0, even(s(s(0)))^2, \ldots \} \]
Definition
Let $P$ be a normal program and $I$ a model. $I$ is a perfect model for a given level of $B_P$, if for every other model $J$, if a positive literal $p$ is the atom of least level in one model, but not in the other, then $p$ is in $J$.

In other words, atoms of higher level are preferred for the perfect model.

Przymusinski: All locally stratified programs have a perfect model, which is independent of the ranking system chosen.