Stratified Programs

People and Knowledge Networks

Observation:

Every normal program is consistent (has a model), but this is not necessarily true for comp(P).

Example:

program P:

$p \leftarrow \sim q$

$q \leftarrow \sim r$

$r \leftarrow \sim p$

I={p,q,r} is a model

comp(P)

$p \leftrightarrow \sim q$

$q \leftrightarrow \sim r$

$r \leftrightarrow \sim p$

$p \leftrightarrow \sim q \leftrightarrow r \leftrightarrow \sim p$

By transitivity:

$p \leftrightarrow \sim p$

Thus there exists no model for comp(P)

Level Mapping

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

Level mapping:
mapping from a set of relation symbols to 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 2 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:

relatedTo(x,y) ← 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)

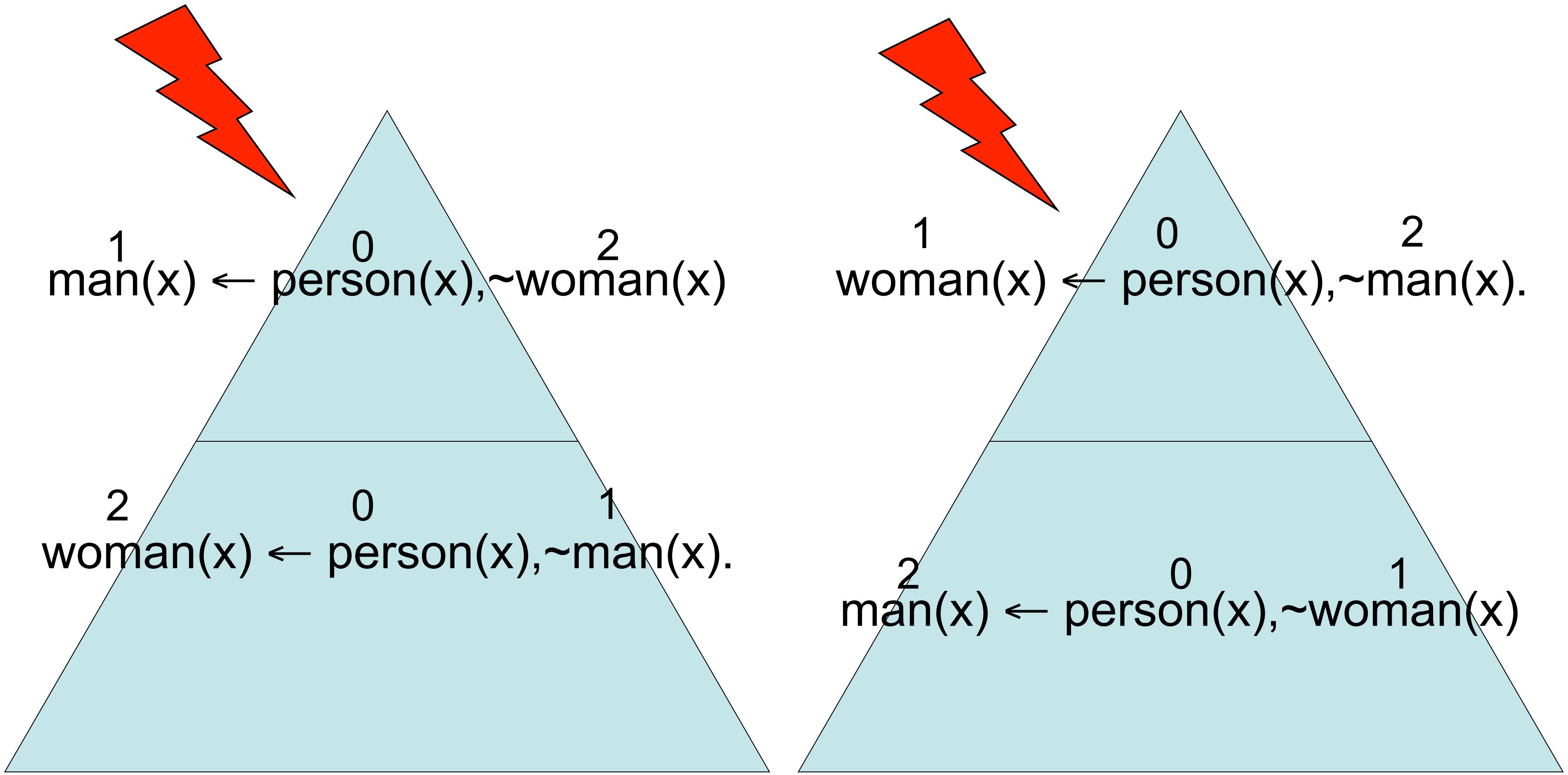
level 1

enemy(x,y) ← ~friend(x,y)
enemy(x,y) <- friend(x,z),enemy(z,y),~friend(x,y)

level 0

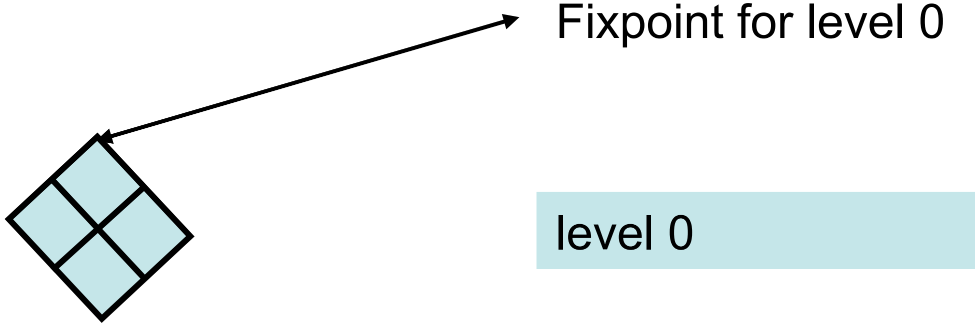
friend(x,z) ← friend(x,y),friend(y,z)
friend(a,b) ←
friend(b,c) ←

Counterexample



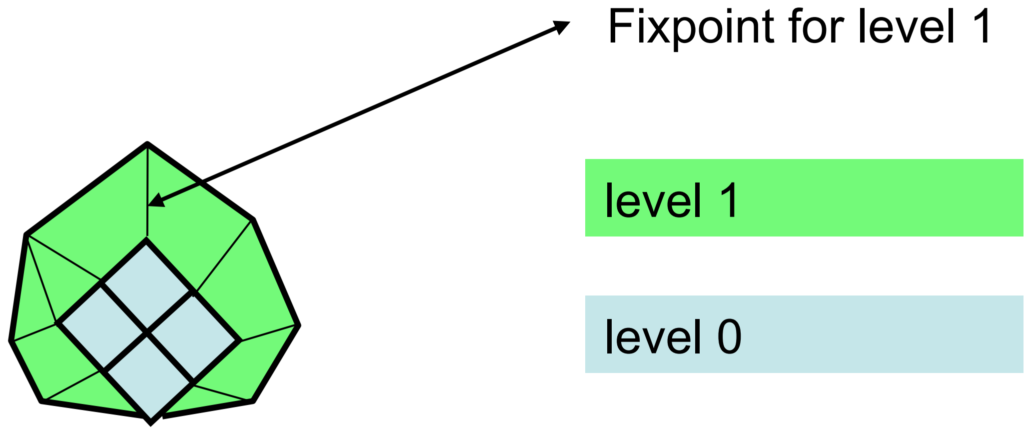
Corollary:

Let P be a stratified normal program. then comp(P) has a minimal normal Herbrand model. A normal Herbrand model assigns the eqality relation to „=".



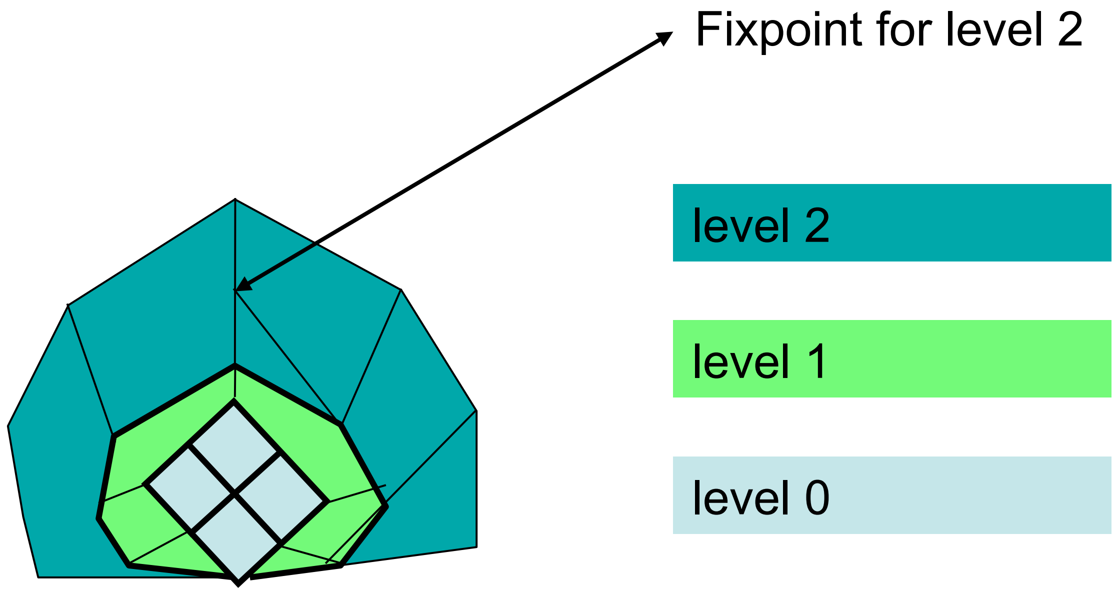
Corollary:

Let P be a stratified normal program. then comp(P) has a minimal normal Herbrand model. A normal Herbrand model assigns the eqality relation to „=”.



Corollary:

Let P be a stratified normal program. then comp(P) has a minimal normal Herbrand model. A normal Herbrand model assigns the eqality relation to „=".



Computational counterpart to models of stratified programmes:

Computing with finite failure

Local Stratification

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 of an atom of greater level
- negatively depends of an atom of equal or greater level.

Example for Local Stratification

$even(s(X)) \leftarrow \neg even(X).$

$even(0).$

B_P:

$\{even(0)^{0}, even(s(0))^{1}, even(s(s(0)))^{2},$
$even(\ldots)^{3}, \ldots\}$

Examples for Local Stratification

$even(s(X)) \leftarrow \neg even(X).$

$even(0).$

$even(0) \leftarrow q(X).$

B_P:

$J=\{q(0)^{\mathbf{0}}, even(0)^{\mathbf{1}}, even(s(s(0)))^{\mathbf{3}}, \ldots\}$

$I=\{even(0)^{\mathbf{0}}, even(s(s(0)))^{\mathbf{2}}, \ldots\}$

Perfect Model

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.