* mp(x) ^ q(y,z) -> r(z)  
  Body = left of the arrow  
  Head = right of the arrow (a.k.a. result)
* A Herbrand *interpretation* is simply a set of ground atoms over a Datalog program. It does not have to include all heads that result from firing rules. The empty set is a valid Herbrand interpretation, as is just the list of facts. It’s a trivial list of ground atoms.
* A Herbrand *model* is a set of ground atoms over a Datalog program where the heads of each rule must be in the set if the body predicates are in the set.
* Bp is the set of all ground atoms (a.k.a. the Herbrand base)
* Tp is the set of all heads (including the facts) created by firing the rules with a given set of inputs
  + A set I is a pre-fixed point of Tp if Tp is a subset of I
  + A set I is a fixed point of Tp if Tp = I
  + Tp can be generated iteratively on successive sets, beginning with the empty set until additional iterations do not increase the size of the set, which yields TP↑ω
* Creating a pre-fixed point set: Supply a list of atoms that, when fired, will result in a set smaller than the supplied list (i.e. add junk atoms that will not fire any rules)
* Creating a fixed point set: Supply a list of atoms that, when fired, will result in the exact same set
* A program Herbrand-*entails* an atom/fact if we can use successive Tp↑ operations to derive that fact. In other words the atom in question is an element of Tp↑⍵.

Information For Exam:

Tp up 0 always is null.

Tp up 1 are always the facts.

When dealing with finding Tp({r(a, d)}) as an example, you have all the facts and then what outputs that are accepted by the specific rule from the given list.

Every prefix point is a herbert model. They are both identical.

In class Examples For Exam Review:

Facts: p(a, b), p(b, c), q(r, a), p(d, d)

Rules:

* p(x, y) -> r(x, y)
* g(x, y) /\ g(y, z) -> g(x, z)

## In-Class Review

**Program**:

Facts:

1. p(a,b)
2. p(b,c)
3. q(c,a)
4. p(d,d)

Rules:

1. p(x,y) -> q(x,y)
2. q(x,y) ^ q(y,z) -> q(x,z)
3. q(x,y) -> r(x,y)
4. r(x,y) -> r(y,x)
5. r(x,x) -> t(x,x)

**Herbrand interpretations**: any set of ground atoms over the language

1. { q(a,a) }
2. { q(a,b), q(b,b), r(a,a) }
3. Empty set

**Herbrand models**: all facts must be in every Herbrand model. Can be found by calculating Tp↑⍵. Tp↑⍵ is the smallest possible Herbrand model

1. Bp (the set of all possible ground atoms in the language)
2. Tp↑⍵
3. Tp↑⍵ ∪ { r(d,d), r(d,a) }  
   Adds r(d,d) because it’s not in the set and r(d,a) based on firing rule 4 with the new input

**Herbrand-entails**: In order to be Herbrand-entailed, the value must exist in Tp↑⍵

1. Does P Herbrand entail r(b,c) ? Yes, because r(b,c) is in Tp↑⍵
2. Does P Herbrand entail r(a,d) ? No, because it’s not

**Computing Tp:** Calculate Tp↑n one single time on the given input (i.e. one single round of the Tp↑⍵ calculation). Tp always contains all the facts, plus rules that can be fired by the given input

1. Tp( { r(a,d) } ) = { p(a,b), p(b,c), q(c,a), p(d,d), r(d,a) }
2. Tp(Bp) { t(x), r(x,y), q(x,y), p(a,b), p(b,c), p(d,d) | x ∈ {a,b,c,d} }  
   It’s the heads of every rule with every possible input  
   In the example, we don’t get every possible p(x,y) because there are no rules that result in p(x,y). We just have p(a,b), p(b,c), and p(d,d) because they are facts

**Pre-fixed points:** Pre-fixed points are exactly the Herbrand models. Therefore, just give Herbrand models

**Fixed points**: Tp↑⍵ is always a fixed point, so just give that (Tp↑⍵ is actually both a fixed and pre-fixed point)