Negated Occurrences of Predicates in PDDL Axiom Bodies

Claudia Grundke and **Gabi Röger**

University of Basel

KRPlan Workshop, ICAPS/KR 2025, Melbourne

Theoretical Starting Point

Axioms in PDDL

```
(:derived (path ?x ?y)
  (or
    (E ?x ?y)
    (exists (?z)
      (and (E ?x ?z) (path ?z ?y)))))

(:derived (acyclic)
  (forall (?x) (not (path ?x ?x))))
```

$$path(x, y) \leftarrow E(x, y) \vee \exists z(E(x, z) \wedge path(z, y))$$
$$acyclic() \leftarrow \forall x \neg path(x, x)$$

Axiom $P(\vec{x}) \leftarrow \varphi(\vec{x})$

- Head P: derived predicate
- Body $\varphi(\vec{x})$: first-order formula

Axioms in PDDL

```
(:derived (path ?x ?y)
  (or
    (E ?x ?y)
    (exists (?z)
      (and (E ?x ?z) (path ?z ?y)))))

(:derived (acyclic)
  (forall (?x) (not (path ?x ?x))))
```

$$path(x, y) \leftarrow E(x, y) \vee \exists z(E(x, z) \wedge path(z, y))$$
$$acyclic() \leftarrow \forall x \neg path(x, x)$$

Axiom $P(\vec{x}) \leftarrow \varphi(\vec{x})$

- Head P: derived predicate
- Body $\varphi(\vec{x})$: first-order formula

An occurrence of derived predicate Q in the body is negative if it is under the scope of an odd number of negations.

Otherwise it's positive.

Axioms in PDDL

```
(:derived (path ?x ?y)
  (or
    (E ?x ?y)
    (exists (?z)
      (and (E ?x ?z) (path ?z ?y)))))

(:derived (acyclic)
  (forall (?x) (not (path ?x ?x))))
```

$$path(x, y) \leftarrow E(x, y) \vee \exists z(E(x, z) \wedge path(z, y))$$
$$acyclic() \leftarrow \forall x \neg path(x, x)$$

Axiom $P(\vec{x}) \leftarrow \varphi(\vec{x})$

- Head P: derived predicate
- Body $\varphi(\vec{x})$: first-order formula

An occurrence of derived predicate Q in the body is negative if it is under the scope of an odd number of negations.

Otherwise it's positive.

Stratification

Stratified axiom program:

sequence $(\Pi_1, \ldots, \Pi_n)$ of disjoint strata
(= finite sets of axioms)

- for all axioms $P(\vec{x}) \leftarrow \varphi(\vec{x})$ in stratum Π_i:
  - if derived predicate P' appears positively in $\varphi(\vec{x})$ then does not not occur as head in later strata.
  - if P' appears negatively in $\varphi(\vec{x})$ then it only occurs as head in strictly earlier strata.

Stratification

Stratified axiom program:

sequence $(\Pi_1, \ldots, \Pi_n)$ of disjoint strata
(= finite sets of axioms)

- for all axioms $P(\vec{x}) \leftarrow \varphi(\vec{x})$ in stratum Π_i:
  - if derived predicate P' appears positively in $\varphi(\vec{x})$ then does not not occur as head in later strata.
  - if P' appears negatively in $\varphi(\vec{x})$ then it only occurs as head in strictly earlier strata.

```
def extend(program : stratified axiom program,
           basic_state, objects):
    s = ... # given assignment from basic_state,
            # and all derived atoms false
    for stratum in program:
        while (there is an instantiation
                of an axiom in stratum such that
                the body is true under s but
                s[A] is false for the head A):
            choose such an A
            s[A] = true
    return s
```

What is Allowed in PDDL?

Thiébaux, Hoffmann, Nebel. *In Defense of PDDL Axioms* (AIJ, 2005)

- **allows any stratifiable set of axioms.**

Edelkamp & Hoffmann, *PDDL2.2: The Language for the Classical Part of the 4th International Planning Competition* (2004, Techn. Report Univ. Freiburg)

- **forbids negative occurrences of derived predicates in axioms bodies**

What is Allowed in PDDL?

Thiébaux, Hoffmann, Nebel. *In Defense of PDDL Axioms* (AIJ, 2005)

- **allows any stratifiable set of axioms.**

Edelkamp & Hoffmann, *PDDL2.2: The Language for the Classical Part of the 4th International Planning Competition* (2004, Techn. Report Univ. Freiburg)

- **forbids negative occurrences of derived predicates in axioms bodies**

This does not make a difference in terms of what we can express.

Both are equivalent to least fixed-point logic on finite structures. (Grundke & Röger, ICAPS 24)

How can we Eliminate Negative Occurrences?

Standing on the Shoulders...

Prior work for least fixed point logics:

- Moschovakis (1974): transformation for infinite structures
- Immerman (1986) and Gurevich (1986): adaptation to finite structures

  We follow the structure by Leivant (1990), notation based on Libkin (2004)

Standing on the Shoulders…

Prior work for least fixed point logics:

- Moschovakis (1974): transformation for infinite structures
- Immerman (1986) and Gurevich (1986): adaptation to finite structures

  We follow the structure by Leivant (1990), notation based on Libkin (2004)

Our contribution:

- adaptation to PDDL axioms
- especially: extension to simultaneous fixed point within each stratum

Stages

Split fixed-point computation for each stratum into several stages.

```
def extend(program : stratified axiom program,
           basic_state, objects):
    s = ... # given assignment from basic_state,
            # and all derived atoms false
    for stratum in program:
        while (there is an instantiation of an axiom in stratum
               such that the body is true under s but
               s[A] is false for the head A):
            choose such an A
            s[A] = true
    return s
```

Stages

Split fixed-point computation for each stratum into several stages.

```
def extend(program : stratified axiom program,
           basic_state, objects):
    s = ... # given assignment from basic_state,
            # and all derived atoms false
    for stratum in program:
        while True:
            # the iterations of this loop correspond to the stages
            s_snapshot = copy of s
            while (there is an instantiation of an axiom in stratum
                   such that the body is true under s_snapshot but
                   s[A] is false for the head A):
                choose such an A
                s[A] = true
            if s == s_snapshot: # no new atoms derived
                break
    return s
```

Stages

Split fixed-point computation for each stratum into several stages.

```
def extend(program : stratified axiom program,
           basic_state, objects):
    s = ... # given assignment from basic_state,
            # and all derived atoms false
    for stratum in program:
        while True:
            # the iterations of this loop correspond to the stages
            s_snapshot = copy of s
            while (there is an instantiation of an axiom in stratum
                   such that the body is true under s_snapshot but
                   s[A] is false for the head A):
                choose such an A
                s[A] = true
            if s == s_snapshot: # no new atoms derived
                break
    return s
```

Stages

Split fixed-point computation for each stratum into several stages.

```
def extend(program : stratified axiom program,
           basic_state, objects):
    s = ... # given assignment from basic_state,
            # and all derived atoms false
    for stratum in program:
        while True:
            # the iterations of this loop correspond to the stages
            s_snapshot = copy of s
            while (there is an instantiation of an axiom in stratum
                   such that the body is true under s_snapshot but
                   s[A] is false for the head A):
                choose such an A
                s[A] = true
            if s == s_snapshot: # no new atoms derived
                break
    return s
```

Stages

Split fixed-point computation for each stratum into several stages.

```
def extend(program : stratified axiom program,
           basic_state, objects):
    s = ... # given assignment from basic_state,
            # and all derived atoms false
    for stratum in program:
        while True:
            # the iterations of this loop correspond to the stages
            s_snapshot = copy of s
            while (there is an instantiation of an axiom in stratum
                   such that the body is true under s_snapshot but
                   s[A] is false for the head A):
                choose such an A
                s[A] = true
            if s == s_snapshot: # no new atoms derived
                break
    return s
```

Stages

Split fixed-point computation for each stratum into several stages.

```
def extend(program : stratified axiom program,
           basic_state, objects):
    s = ... # given assignment from basic_state,
            # and all derived atoms false
    for stratum in program:
        while True:
            # the iterations of this loop correspond to the stages
            s_snapshot = copy of s
            while (there is an instantiation of an axiom in stratum
                   such that the body is true under s_snapshot but
                   s[A] is false for the head A):
                choose such an A
                s[A] = true
            if s == s_snapshot: # no new atoms derived
                break
    return s
```

Stage Relations

Consider the stages of a single stratum.

Stage Relations

Consider the stages of a single stratum.

- f: index of the last stage (first reaching the fixed point)

Stage Relations

Consider the stages of a single stratum.

- f: index of the last stage (first reaching the fixed point)
- $|\vec{a}|_P$: index of the stage in which $P(\vec{a})$ is made true or $f+1$ if this is never the case

Stage Relations

Consider the stages of a single stratum.

- f: index of the last stage (first reaching the fixed point)
- $|\vec{a}|_P$: index of the stage in which $P(\vec{a})$ is made true or $f+1$ if this is never the case
- $\vec{a} \prec^{P,Q} \vec{b}$ iff $|\vec{a}|_P < |\vec{b}|_Q$

  $P(\vec{a})$ is derived at a strictly earlier stage than $Q(\vec{b})$"

Stage Relations

Consider the stages of a single stratum.

- f: index of the last stage (first reaching the fixed point)
- $|\vec{a}|_P$: index of the stage in which $P(\vec{a})$ is made true or $f+1$ if this is never the case
- $\vec{a} \prec^{P,Q} \vec{b}$ iff $|\vec{a}|_P < |\vec{b}|_Q$

  $P(\vec{a})$ is derived at a strictly earlier stage than $Q(\vec{b})$"
- $\vec{a} \preceq^{P,Q} \vec{b}$ iff $|\vec{a}|_P \leq |\vec{b}|_Q$ and $|\vec{a}|_P \leq f$

  "$P(\vec{a})$ is derived and at an earlier stage than $Q(\vec{b})$"

Stage Relations

Consider the stages of a single stratum.

- f: index of the last stage (first reaching the fixed point)
- $|\vec{a}|_P$: index of the stage in which $P(\vec{a})$ is made true or $f+1$ if this is never the case
- $\vec{a} \prec^{P,Q} \vec{b}$ iff $|\vec{a}|_P < |\vec{b}|_Q$
  $P(\vec{a})$ is derived at a strictly earlier stage than $Q(\vec{b})$"
- $\vec{a} \preceq^{P,Q} \vec{b}$ iff $|\vec{a}|_P \leq |\vec{b}|_Q$ and $|\vec{a}|_P \leq f$
  "$P(\vec{a})$ is derived and at an earlier stage than $Q(\vec{b})$"
- $\vec{a} \lhd^{P,Q} \vec{b}$ iff $|\vec{a}|_P + 1 = |\vec{b}|_Q$
  "$P(\vec{a})$ is derived in the predecessor stage of $Q(\vec{b})$"

Stage Relations

Consider the stages of a single stratum.

- f: index of the last stage (first reaching the fixed point)
- $|\vec{a}|_P$: index of the stage in which $P(\vec{a})$ is made true or $f+1$ if this is never the case
- $\vec{a} \prec^{P,Q} \vec{b}$ iff $|\vec{a}|_P < |\vec{b}|_Q$

  $P(\vec{a})$ is derived at a strictly earlier stage than $Q(\vec{b})$"
- $\vec{a} \preceq^{P,Q} \vec{b}$ iff $|\vec{a}|_P \leq |\vec{b}|_Q$ and $|\vec{a}|_P \leq f$

  "$P(\vec{a})$ is derived and at an earlier stage than $Q(\vec{b})$"
- $\vec{a} \lhd^{P,Q} \vec{b}$ iff $|\vec{a}|_P + 1 = |\vec{b}|_Q$

  "$P(\vec{a})$ is derived in the predecessor stage of $Q(\vec{b})$"
- $\not\prec^{P,Q}$ and $\not\preceq^{P,Q}$ for complement of $\prec^{P,Q}$ and $\preceq^{P,Q}$

Elimination

Elimination

- The stage relations can be expressed by axioms without negated occurrences of derived predicates.

Elimination

- The stage relations can be expressed by axioms without negated occurrences of derived predicates.
- $\vec{x} \preceq^{P,P} \vec{x}$ and $\neg\vec{x} \not\preceq^{P,P} \vec{x}$ are true iff $P(\vec{x})$ is true in the fixed point

Elimination

- The stage relations can be expressed by axioms without negated occurrences of derived predicates.
- $\vec{x} \preceq^{P,P} \vec{x}$ and $\neg\vec{x} \not\preceq^{P,P} \vec{x}$ are true iff $P(\vec{x})$ is true in the fixed point
- Idea: Replace negated occurrences of $P(\vec{x})$ in axiom bodies with $\neg\vec{x} \not\preceq^{P,P} \vec{x}$

Elimination

- The stage relations can be expressed by axioms without negated occurrences of derived predicates.
- $\vec{x} \preceq^{P,P} \vec{x}$ and $\neg\vec{x} \not\preceq^{P,P} \vec{x}$ are true iff $P(\vec{x})$ is true in the fixed point
- Idea: Replace negated occurrences of $P(\vec{x})$ in axiom bodies with $\neg\vec{x} \not\preceq^{P,P} \vec{x}$

for Π_ℓ in $\Pi_1, \ldots, \Pi_{n-1}$ **do**
  affected $:= \{P \mid P$ is affected by an axiom from $\Pi_\ell\}$
  if no predicate from *affected* occurs negatively in any axiom **then**
    continue
  add axioms that express the stage axioms for Π_ℓ to Π_ℓ.
  for Π in $\Pi_{\ell+1}, \ldots, \Pi_n$ **do**
    for each axiom ax in Π **do**
      replace in ax all negative occurrences of some $P_i(\vec{x})$
      where $P_i \in$ *affected* with $\neg\vec{x} \not\preceq^{i,i} \vec{x}$

How can we Express the Stage Relations with Axioms?

Notation

- $P_1, \ldots, P_m$: predicates derived on the processed stratum
- W.l.o.g. only a single axiom $P_i(\vec{x}) \leftarrow \varphi_i(\vec{x})$ for each such predicate.
- Write $\prec^{i,j}$ for $\prec^{P_i,P_j}$

Notation

- $P_1, \ldots, P_m$: predicates derived on the processed stratum
- W.l.o.g. only a single axiom $P_i(\vec{x}) \leftarrow \varphi_i(\vec{x})$ for each such predicate.
- Write $\prec^{i,j}$ for $\prec^{P_i,P_j}$

- $\varphi(\vec{x})[\prec^j \vec{y}]$: replace every occurrence of an atom $P_k(\vec{z})$ with $k \in \{1, \ldots, m\}$ in $\varphi(\vec{x})$ by $\vec{z} \prec^{k,j} \vec{y}$
  Like $\varphi(\vec{x})$ but in the evaluation we may only use the atoms derived in a stage strictly before $P_j(\vec{y})$.

Notation

- $P_1, \ldots, P_m$: predicates derived on the processed stratum
- W.l.o.g. only a single axiom $P_i(\vec{x}) \leftarrow \varphi_i(\vec{x})$ for each such predicate.
- Write $\prec^{i,j}$ for $\prec^{P_i,P_j}$

- $\varphi(\vec{x})[\prec^j \vec{y}]$: replace every occurrence of an atom $P_k(\vec{z})$ with $k \in \{1, \ldots, m\}$ in $\varphi(\vec{x})$ by $\vec{z} \prec^{k,j} \vec{y}$
  Like $\varphi(\vec{x})$ but in the evaluation we may only use the atoms derived in a stage strictly before $P_j(\vec{y})$.
- analogously $\varphi(\vec{x})[\preceq^j \vec{y}]$

Notation

- $P_1, \ldots, P_m$: predicates derived on the processed stratum
- W.l.o.g. only a single axiom $P_i(\vec{x}) \leftarrow \varphi_i(\vec{x})$ for each such predicate.
- Write $\prec^{i,j}$ for $\prec^{P_i,P_j}$

- $\varphi(\vec{x})[\prec^j \vec{y}]$: replace every occurrence of an atom $P_k(\vec{z})$ with $k \in \{1, \ldots, m\}$ in $\varphi(\vec{x})$ by $\vec{z} \prec^{k,j} \vec{y}$
  Like $\varphi(\vec{x})$ but in the evaluation we may only use the atoms derived in a stage strictly before $P_j(\vec{y})$.
- analogously $\varphi(\vec{x})[\preceq^j \vec{y}]$
- $\varphi(\vec{x})[\neg \not\prec^j \vec{y}]$: replace every occurrence of an atom $P_k(\vec{z})$ with $k \in \{1, \ldots, m\}$ in $\varphi(\vec{x})$ by $\neg\vec{z} \not\prec^{k,j} \vec{y}$
  Same idea. Used if we need to negate the result.

Notation

- $P_1, \ldots, P_m$: predicates derived on the processed stratum
- W.l.o.g. only a single axiom $P_i(\vec{x}) \leftarrow \varphi_i(\vec{x})$ for each such predicate.
- Write $\prec^{i,j}$ for $\prec^{P_i,P_j}$

- $\varphi(\vec{x})[\prec^j \vec{y}]$: replace every occurrence of an atom $P_k(\vec{z})$ with $k \in \{1, \ldots, m\}$ in $\varphi(\vec{x})$ by $\vec{z} \prec^{k,j} \vec{y}$
  Like $\varphi(\vec{x})$ but in the evaluation we may only use the atoms derived in a stage strictly before $P_j(\vec{y})$.
- analogously $\varphi(\vec{x})[\preceq^j \vec{y}]$
- $\varphi(\vec{x})[\neg \not\prec^j \vec{y}]$: replace every occurrence of an atom $P_k(\vec{z})$ with $k \in \{1, \ldots, m\}$ in $\varphi(\vec{x})$ by $\neg \vec{z} \not\prec^{k,j} \vec{y}$
  Same idea. Used if we need to negate the result.
- $\varphi(\vec{x})[\bot]$: replace every occurrence of an atom $P_k(\vec{z})$ with $\bot$
  True if $\varphi(\vec{x})$ is already made true in the first stage.

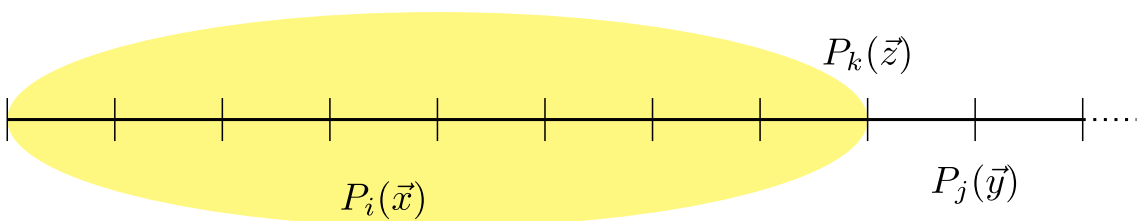
$$\vec{x} \prec^{i,j} \vec{y} \leftarrow \bigvee_{k=1}^{m} \exists \vec{z} (\vec{x} \preceq^{i,k} \vec{z} \wedge \vec{z} \lhd^{k,j} \vec{y})$$

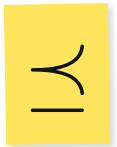
$P_i(\vec{x})$ is derived strictly before $P_j(\vec{y})$
if it is derived before some $P_k(\vec{z})$ that is derived immediately before $P_j(\vec{y})$.

$\prec$

$\vec{x} \prec^{i,j} \vec{y} \leftarrow \bigvee_{k=1}^{m} \exists \vec{z} (\vec{x} \preceq^{i,k} \vec{z} \wedge \vec{z} \lhd^{k,j} \vec{y})$

$P_i(\vec{x})$ is derived strictly before $P_j(\vec{y})$
if it is derived before some $P_k(\vec{z})$ that is derived immediately before $P_j(\vec{y})$.





$\vec{x} \preceq^{i,j} \vec{y} \leftarrow \varphi_i(\vec{x})[\prec^j \vec{y}]$

$P_i(\vec{x})$ is derived before $P_j(\vec{y})$
because $P_i(\vec{x})$ can already be derived using only atoms that are derived strictly before $P_j(\vec{y})$.

$\prec$

$\vec{x} \preceq^{i,j} \vec{y} \leftarrow \varphi_i(\vec{x})[\prec^j \vec{y}]$

$P_i(\vec{x})$ is derived before $P_j(\vec{y})$
because $P_i(\vec{x})$ can already be derived using only atoms that are derived strictly before $P_j(\vec{y})$.

$$\vec{x} \not\prec^{i,j} \vec{y} \leftarrow \varphi_j(\vec{y})[\bot] \vee \left(\bigvee_{k=1}^{m} \exists \vec{z} (\vec{x} \not\prec^{i,k} \vec{z} \wedge \vec{z} \lhd^{k,j} \vec{y}) \right) \vee \left(\bigwedge_{k=1}^{m} \forall \vec{z} \neg \varphi_k(\vec{z})[\bot] \right)$$

$P_i(\vec{x})$ is not derived strictly before $P_j(\vec{y})$ because

- $P_j(\vec{y})$ is already derived in stage 1,
- there is some $P_j(\vec{z})$ derived immediately before $P_j(\vec{y})$ and $P_i(\vec{x})$ is not derived before this $P_k(\vec{z})$, so $P_i(\vec{x})$ is not derived strictly before $P_j(\vec{y})$, or
- nothing can be derived at all, so both atoms are in the same (last) stage.

$$\vec{x} \not\prec^{i,j} \vec{y} \leftarrow \varphi_j(\vec{y})[\bot] \vee \left(\bigvee_{k=1}^{m} \exists \vec{z} (\vec{x} \not\prec^{i,k} \vec{z} \wedge \vec{z} \lhd^{k,j} \vec{y}) \right) \vee \left(\bigwedge_{k=1}^{m} \forall \vec{z} \neg \varphi_k(\vec{z})[\bot] \right)$$

$P_i(\vec{x})$ is not derived strictly before $P_j(\vec{y})$ because

- $P_j(\vec{y})$ is already derived in stage 1,
- there is some $P_j(\vec{z})$ derived immediately before $P_j(\vec{y})$ and $P_i(\vec{x})$ is not derived before this $P_k(\vec{z})$, so $P_i(\vec{x})$ is not derived strictly before $P_j(\vec{y})$, or
- nothing can be derived at all, so both atoms are in the same (last) stage.

$\vec{x} \not\preceq^{i,j} \vec{y} \leftarrow \neg\varphi_i(\vec{x})[\neg \not\prec^j \vec{y}]$

$P_i(\vec{x})$ is not derived before $P_j(\vec{y})$
because the atoms derived strictly before $P_j(\vec{y})$ are not sufficient to derive it
(expressing $\prec$ as negated $\not\prec$ to avoid a negated occurrence of $\prec$ in the overall negated formula).

$\not\prec$

$\vec{x} \not\prec^{i,j} \vec{y} \leftarrow \neg\varphi_i(\vec{x})[\neg \not\prec^j \vec{y}]$

$P_i(\vec{x})$ is not derived before $P_j(\vec{y})$
because the atoms derived strictly before $P_j(\vec{y})$ are not sufficient to derive it
(expressing $\prec$ as negated $\not\prec$ to avoid a negated occurrence of $\prec$ in the overall negated formula).

$$\vec{x} \lhd^{i,j} \vec{y} \leftarrow \varphi_i(\vec{x})[\prec^i \vec{x}] \wedge \neg\varphi_j(\vec{y})[\neg \not\prec^i \vec{x}] \wedge$$

$$\left(\varphi_j(\vec{y})[\preceq^i \vec{x}] \vee \bigwedge_{k=1}^{m} \forall\vec{z}(\neg\varphi_k(\vec{z})[\neg \not\preceq^i \vec{x}] \vee \varphi_k(\vec{z})[\prec^i \vec{x}])\right)$$

Three requirements for $P_i(\vec{x})$ being derived immediately before $P_j(\vec{y})$:

- $P_i(\vec{x})$ can be derived from the atoms derived strictly before $P_i(\vec{x})$, so it is true in the fixed point,
- $P_j(\vec{y})$ cannot be derived from the atoms derived strictly before $P_i(\vec{x})$, so it is not derived at the same stage as $P_i(\vec{x})$ (or earlier), and
- $P_j(\vec{y})$ can be derived from the atoms derived before $P_i(\vec{x})$, or $P_i(\vec{x})$ was derived in the stage that reached the fixed point.

$$\vec{x} \lhd^{i,j} \vec{y} \leftarrow \varphi_i(\vec{x})[\prec^i \vec{x}] \wedge \neg\varphi_j(\vec{y})[\neg \not\prec^i \vec{x}] \wedge$$
$$\left(\varphi_j(\vec{y})[\preceq^i \vec{x}] \vee \bigwedge_{k=1}^{m} \forall\vec{z}(\neg\varphi_k(\vec{z})[\neg \not\preceq^i \vec{x}] \vee \varphi_k(\vec{z})[\prec^i \vec{x}])\right)$$

Three requirements for $P_i(\vec{x})$ being derived immediately before $P_j(\vec{y})$:

- $P_i(\vec{x})$ can be derived from the atoms derived strictly before $P_i(\vec{x})$, so it is true in the fixed point,
- $P_j(\vec{y})$ cannot be derived from the atoms derived strictly before $P_i(\vec{x})$, so it is not derived at the same stage as $P_i(\vec{x})$ (or earlier), and
- $P_j(\vec{y})$ can be derived from the atoms derived before $P_i(\vec{x})$, or $P_i(\vec{x})$ was derived in the stage that reached the fixed point.

Summary and Future Work

- We can eliminate negative occurrences of derived predicates from axiom bodies.
- Main idea: make the derivation order explicit (to the level of stages).
- This allows us to positively express that an atom is not derived by the original axioms.
- In future work, we will empirically evaluate the transformation.