Effiziente Auswertung von linearen Datalog−Programmen (ohne Negation)

Vereinfachung der Darstellung:
Sei P ein rekursives Kopfprädikat mit
genau einer nichtrekursiven Regel $Q1(...) \wedge ... \wedge Qm(...) \Rightarrow P(...)$ und
genau einer linearen Regel $R1(...) \wedge ... \wedge Rn(...) \wedge P(...) \Rightarrow P(...)$.

Naive Bottom−up−Auswertung:

$P_0 :=$ Auswertung des Rumpfes $Q1(...) \wedge ... \wedge Qm(...)$
$i := 0$
repeat
  $i := i+1$
  $P_i :=$ Auswertung des Rumpfes $R1(...) \wedge ... \wedge Rn(...) \wedge P_{i-1}(...)$ /* Join */
  $P_i := P_i \cup P_{i-1}$
until $P_i = P_{i-1}$ /* Erreichung eines Fixpunkts */
$P := P_i$

Semi−naive Bottom−up−Auswertung (Delta−Iteration):

$P_0 :=$ Auswertung des Rumpfes $Q1(...) \wedge ... \wedge Qm(...)$
$\Delta_0 := P_0$
$i := 0$
repeat
  $i := i+1$
  $\Delta_i :=$ Auswertung des Rumpfes $R1(...) \wedge ... \wedge Rn(...) \wedge \Delta_{i-1}(...)$ /* Join */
  $\Delta_i := \Delta_i - P_{i-1}$
  $P_i := P_{i-1} \cup \Delta_i$
until $\Delta_i = \emptyset$ /* Erreichung eines Fixpunkts */
$P := P_i$

Beispiel:

Flüge (a,z) ⇒ Flugverbindungen (a,z)
Flugverbindungen (a,o) ∧ Flüge (o,z) ⇒ Flugverbindungen (a,z)
Anfrage:
  Flugverbindungen (Frankfurt, z)

Naive Auswertung:

i	P_i
0	(F,C), (C,P), (P,Y), (F,D), (D,P), (C,D)
1	(F,C), (C,P), (P,Y), (F,D), (D,P), (C,D), (F,P), (C,Y), (D,Y)
2	(F,C), (C,P), (P,Y), (F,D), (D,P), (C,D), (F,P), (C,Y), (D,Y), (F,Y)
3	(F,C), (C,P), (P,Y), (F,D), (D,P), (C,D), (F,P), (C,Y), (D,Y), (F,Y)

Semi−naive Auswertung:

i	P_i	Δ_i
0	(F,C), (C,P), (P,Y), (F,D), (D,P), (C,D)	(F,C), (C,P), (P,Y), (F,D), (D,P), (C,D)
1	(F,C), (C,P), (P,Y), (F,D), (D,P), (C,D), (F,P), (C,Y), (D,Y)	(F,P), (C,Y), (D,Y)
2	(F,C), (C,P), (P,Y), (F,D), (D,P), (C,D), (F,P), (C,Y), (D,Y), (F,Y)	(F,Y)
3	(F,C), (C,P), (P,Y), (F,D), (D,P), (C,D), (F,P), (C,Y), (D,Y), (F,Y)	−