Tabling with Sound Answer Subsumption

Alexander Vandenbroucke

Maciej Piróg - Benoit Desouter - Tom Schrijvers

KU LEUVEN

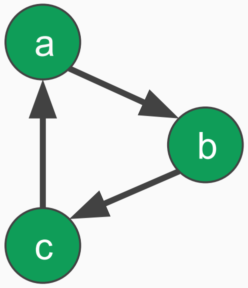
UNIVERSITEIT GENT

Tabling

Regular Prolog

```
edge(a,b). edge(b,c). edge(c,a).

path(X,Y) :- edge(X,Y).
path(X,Y) :- edge(X,Z),path(Z,Y).
```

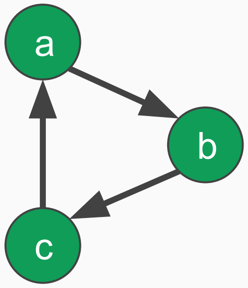


Regular Prolog

```
edge(a,b). edge(b,c). edge(c,a).

path(X,Y) :- edge(X,Y).
path(X,Y) :- edge(X,Z),path(Z,Y).
```

?- path(a,b).

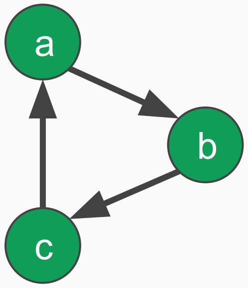


Regular Prolog

```
edge(a,b). edge(b,c). edge(c,a).

path(X,Y) :- edge(X,Y).
path(X,Y) :- edge(X,Z),path(Z,Y).
```

?- path(a,b).
true.

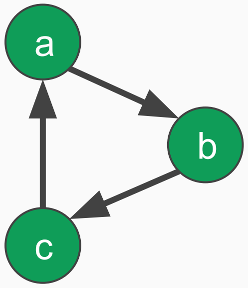


Regular Prolog

```
edge(a,b). edge(b,c). edge(c,a).

path(X,Y) :- edge(X,Y).
path(X,Y) :- edge(X,Z),path(Z,Y).
```

?- path(a,b).
true;
true;
...

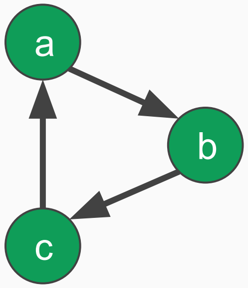


Regular Prolog

```
edge(a,b). edge(b,c). edge(c,a).

path(X,Y) :- edge(X,Y).
path(X,Y) :- edge(X,Z),path(Z,Y).
```

?- path(a,d).

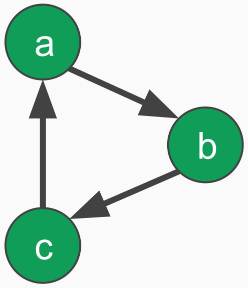


Regular Prolog

```
edge(a,b). edge(b,c). edge(c,a).

path(X,Y) :- edge(X,Y).
path(X,Y) :- edge(X,Z),path(Z,Y).
```

?- path(a,d).
<infinite loop>



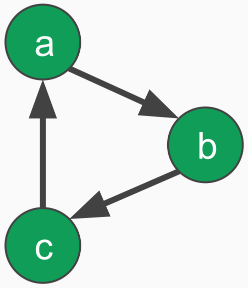
Tabled Prolog

```
:- table path/2.
edge(a,b). edge(b,c). edge(c,a).

path(X,Y) :- edge(X,Y).
path(X,Y) :- edge(X,Z),path(Z,Y).
```

?- path(a,b).
true.

?- path(a,d).
false.



Tabled Prolog

YAP!

Applied Logic Systems

SWI Prolog

PROLOG B

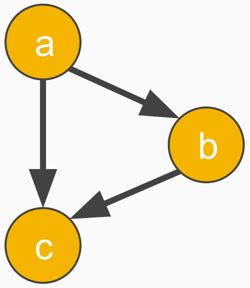
xsb

Aggregation

Regular Prolog

```
p(X,Y,1) :- e(X,Y).
p(X,Y,D) :- e(X,Z), p(Z,Y,D1),
            D is D1 + 1.
shortest(X,Y,MinD) :-
  findall(D,p(X,Y,D),List),
  min_list(List,MinD).
```

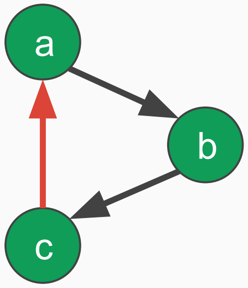
?- shortest(a,c,D).
D = 1.



Regular Prolog

```
p(X,Y,1) :- e(X,Y).
p(X,Y,D) :- e(X,Z), p(Z,Y,D1),
            D is D1 + 1.
shortest(X,Y,MinD) :-
  findall(D,p(X,Y,D),List),
  min_list(List,MinD).
```

?- shortest(a,c,D).
<infinite loop>

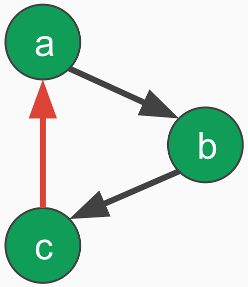


Answer Subsumption

```
:- table p/3.

p(X,Y,1) :- e(X,Y).
p(X,Y,D) :- e(X,Z), p(Z,Y,D1),
            D is D1 + 1.
```

?- p(a,c,D).
<infinite loop>

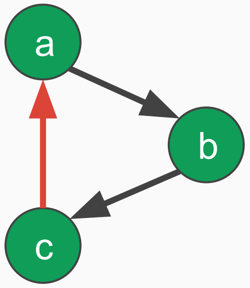


Answer Subsumption

```
:- table p(+,+,min).

p(X,Y,1) :- e(X,Y).
p(X,Y,D) :- e(X,Z), p(Z,Y,D1),
            D is D1 + 1.
```

?- p(a,c,D).
D = 2.



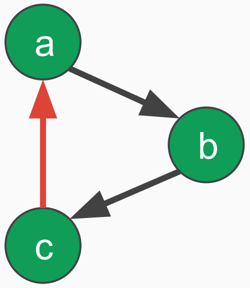
Answer Subsumption

```
:- table p(+,+,min).

p(X,Y,1) :- e(X,Y).
p(X,Y,D) :- e(X,Z), p(Z,Y,D1),
            D is D1 + 1.
```

?- p(a,c,D).
D = 2.

tabling modes



Example 2

```
p(0).
p(1).
p(2) :- p(X), X = 1.
p(3) :- p(X), X = 0.
```

?- p(X).
X = 0;
X = 1;
X = 2;
X = 3.

Example 2

```
:- table p(max).
p(0).
p(1).
p(2) :- p(X), X = 1.
p(3) :- p(X), X = 0.
```

?- p(X).

Example 2

```
:- table p(max).
p(0).
p(1).
p(2) :- p(X), X = 1.
p(3) :- p(X), X = 0.
```

?- p(X).

X = 0;

X = 1;

X = 2.

Example 2

```
:- table p(max).
p(0).
p(1).
p(2) :- p(X), X = 1.
p(3) :- p(X), X = 0.
```

?- p(X).

X = 0;

X = 1;

X = 2.

1 overwrites 0

the last rule does not apply!

Example 2

```
:- table p(max).
p(0).
p(1).
p(2) :- p(X), X = 1.
p(3) :- p(X), X = 0.
```

?- p(X).

X = 0;

X = 1;

X = 2.

Operational semantics

Tabling Semantics

Tabling Semantics

Assumption:
Tabling systems implement
***least-fixed point semantics*.**

Tabling Semantics

Immediate Consequence

$$T_P : \mathcal{P}(\overbrace{H}^{\text{all atoms}}) \rightarrow \mathcal{P}(H)$$

$$T_P\underbrace{(I)}_{\substack{\text{known}\\\text{atoms}}} = \{\underbrace{A}_{\substack{\text{derived}\\\text{atoms}}} \mid A \text{ :- } B1,\ldots,Bn \in ground(P) \wedge \{B1,\ldots,Bn\} \subseteq I\}$$

Tabling Semantics

Immediate Consequence

$$T_P : \mathcal{P}(\overbrace{H}^{\text{all atoms}}) \longrightarrow \mathcal{P}(H)$$

$$T_P(\underbrace{I}_{\substack{\text{known}\\\text{atoms}}}) = \{\underbrace{A}_{\substack{\text{derived}\\\text{atoms}}} \;\mid\; A \text{ :- } B1,\ldots,Bn \in ground(P) \;\wedge\; \{B1,\ldots,Bn\} \subseteq I\}$$

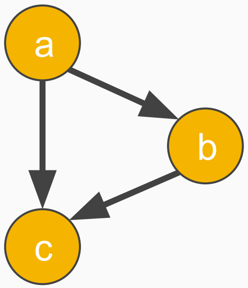
Semantics

$$\boxed{\mathbf{lfp(T_P)}}$$

Example

```
e(a,b). e(b,c). e(a,c).

p(X,Y) :- e(X,Y).
p(X,Y) :- e(X,Z),p(Z,Y).
```

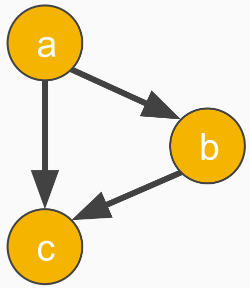


$$I_0 = \varnothing$$

Example

```
e(a,b). e(b,c). e(a,c).

p(X,Y) :- e(X,Y).
p(X,Y) :- e(X,Z),p(Z,Y).
```



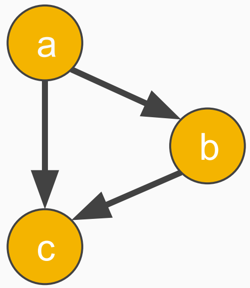
$I_0 = \varnothing$

$I_1 = T_P(I_0) = \{$ ***e(a,b), e(b,c), e(a,c)*** $\}$

Example

```
e(a,b). e(b,c). e(a,c).

p(X,Y) :- e(X,Y).
p(X,Y) :- e(X,Z),p(Z,Y).
```

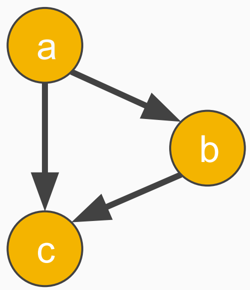


$$I_0 = \varnothing$$
$$I_1 = T_P(I_0) = \{\, e(a,b),\ e(b,c),\ e(a,c) \,\}$$
$$I_2 = T_P(I_1) = \{\, e(a,b),\ e(b,c),\ e(a,c),$$
$$\mathbf{p(a,b),\ p(b,c),\ p(a,c)} \,\}$$

Example

```
e(a,b). e(b,c). e(a,c).

p(X,Y) :- e(X,Y).
p(X,Y) :- e(X,Z),p(Z,Y).
```



$$I_0 = \varnothing$$

$$I_1 = T_P(I_0) = \{\, e(a,b),\ e(b,c),\ e(a,c) \,\}$$

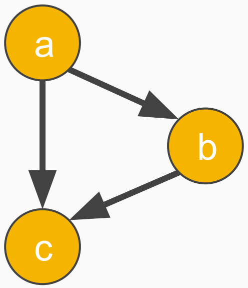
$$I_2 = T_P(I_1) = \{\, e(a,b),\ e(b,c),\ e(a,c),\ p(a,b),\ p(b,c),\ p(a,c) \,\}$$

$$I_3 = T_P(I_2) = \{\, e(a,b),\ e(b,c),\ e(a,c),\ p(a,b),\ p(b,c),\ p(a,c) \,\}$$

Example

```
e(a,b). e(b,c). e(a,c).

p(X,Y) :- e(X,Y).
p(X,Y) :- e(X,Z),p(Z,Y).
```



$I_0 = \varnothing$

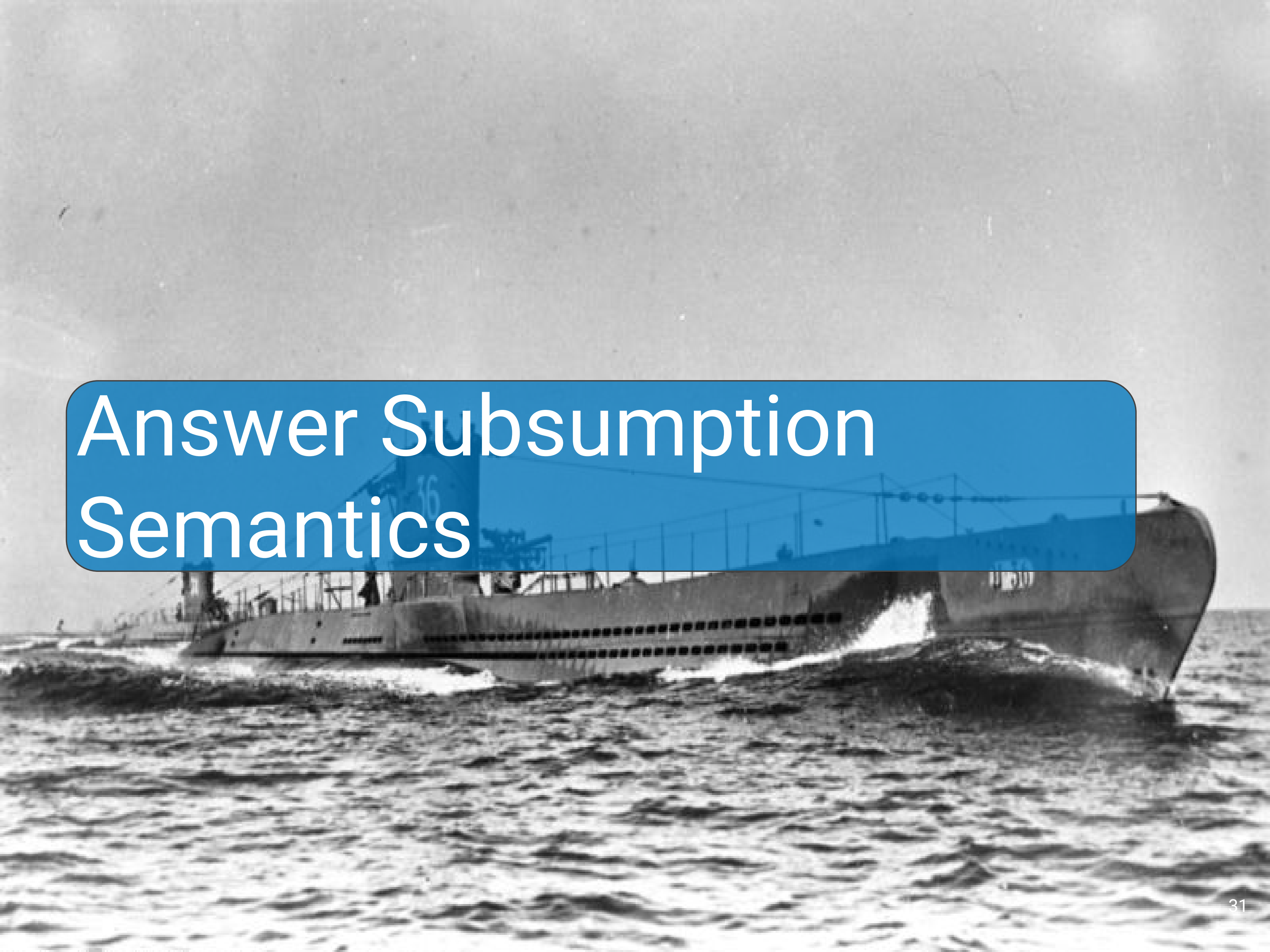
$I_1 = T_P(I_0) = \{\, e(a,b),\ e(b,c),\ e(a,c) \}$

$I_2 = T_P(I_1) = \{\, e(a,b),\ e(b,c),\ e(a,c),$
$p(a,b),\ p(b,c),\ p(a,c) \}$

$I_3 = T_P(I_2) = \{\, e(a,b),\ e(b,c),\ e(a,c),$
$p(a,b),\ p(b,c),\ p(a,c) \}$

$T_P(I_2) = I_2$
least fixpoint

Answer Subsumption Semantics



Answer Subsumption Semantics

Immediate Consequence

$$T_P : \mathcal{P}(H) \longrightarrow \mathcal{P}(H)$$

Tabling Semantics

$$\boxed{\mathbf{lfp(T_P)}}$$

Immediate Consequence

$$T_P : \mathcal{P}(H) \rightarrow \mathcal{P}(H)$$

Post-processing Aggregation

$\alpha : \mathcal{P}(H) \rightarrow L$ in some lattice $\langle L, \leq_L \rangle$

Answer Subsumption Semantics

$$\boxed{\boldsymbol{\alpha}(\mathbf{lfp}(\mathbf{T_P}))}$$

Operational independence - Good

<u>tabling mode</u>

$\alpha =$ `min`

$\alpha =$ `max`

$\alpha =$ `all`

$\alpha =$ `lattice`

provided by

Operational independence - Bad

$\alpha = \texttt{first}$

$\alpha = \texttt{last}$

$\alpha = \texttt{sum}$

provided by

depends on ordering
⇒ Not lattices

Example

Maximum lattice on the argument

$$\langle \{0,1,2,3\}, \geq \rangle$$

$$\alpha(S) = max_{\leq} \{ x \mid p(x) \in S \}$$

general derivation in the paper

Example

```
p(0).
p(1).
p(2) :- p(X), X = 1.
p(3) :- p(X), X = 0.
```

$$\alpha(\mathrm{lfp}(T_P))$$

Example

```
p(0).
p(1).
p(2) :- p(X), X = 1.
p(3) :- p(X), X = 0.
```

$\alpha(\{p(0),\ p(1)\})$

Example

```
p(0).
p(1).
p(2) :- p(X), X = 1.
p(3) :- p(X), X = 0.
```

$$\alpha(\{p(0),\ p(1), p(2), p(3)\})$$

Example

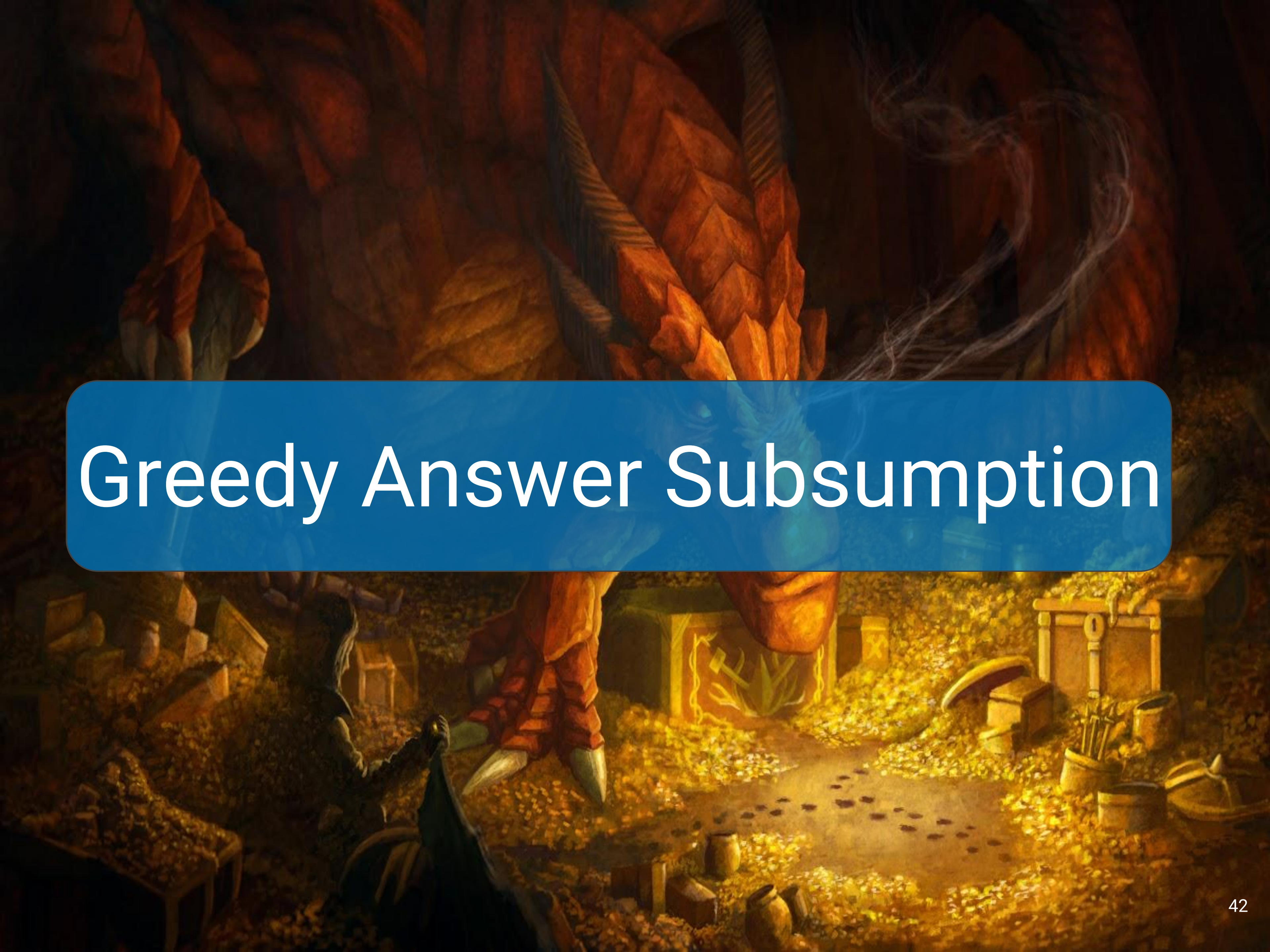
```
p(0).
p(1).
p(2) :- p(X), X = 1.
p(3) :- p(X), X = 0.
```

$$max_{\leq}(\{0, 1, 2, 3\})$$

Example

```
p(0).
p(1).
p(2) :- p(X), X = 1.
p(3) :- p(X), X = 0.
```

{3}



Greedy Answer Subsumption

Generalised Semantics

$$\langle L, \alpha^L, T_P{}^L \rangle$$

$$\boxed{\mathbf{lfp}(T_P{}^L)}$$

Generalised Semantics

$$\langle L, \alpha^L, T_P{}^L \rangle$$

complete lattice

Generalised Semantics

$$\langle L, \alpha^L, T_P{}^L \rangle$$

complete lattice

$\alpha^L : \mathcal{P}(H) \rightarrow L$

mapping function

Generalised Semantics

$$\langle L, \alpha^L, T_P{}^L \rangle$$

complete lattice

$\alpha^L : \mathcal{P}(H) \to L$
mapping function

$T_P{}^L : L \to L$
generalised immediate consequence (monotone)

Generalised Semantics

$$\langle L, \alpha^L, T_P{}^L \rangle$$

complete lattice

$\alpha^L : \mathcal{P}(H) \rightarrow L$
mapping function

$T_P{}^L : L \rightarrow L$
generalised immediate consequence (monotone)

$$\boxed{\mathtt{lfp}(T_P{}^L)}$$

Answer Subsumption

Correctness

$$\mathrm{lfp}(\mathrm{T}_{\mathrm{P}}{}^{\mathrm{L}}) = \alpha^{L}(\mathrm{lfp}(T_P))$$

Answer Subsumption

Correctness

$$\mathrm{lfp}(T_P^L) = \alpha^L(\mathrm{lfp}(T_P))$$

General Correctness Condition

$$\alpha^L \circ T_P = T_P^L \circ \alpha^L$$

Answer Subsumption

Correctness

$$\mathrm{lfp}(T_P^L) = \alpha^L(\mathrm{lfp}(T_P))$$

General Correctness Condition

$$\alpha^L \circ T_P = T_P^L \circ \alpha^L$$

aggregation

immediate consequence

lattice immediate consequence

Specific Generalised Semantics

$$\langle L, \alpha^L, T_P{}^L \rangle$$

$L(\leqslant)$
see paper

$\alpha : \mathcal{P}(H) \rightarrow L(\leqslant)$
see paper

$\alpha \circ T_P \circ \rho$
greedy

$\rho : L(\leqslant) \rightarrow \mathcal{P}(H)$
extract

Answer Subsumption

Specific Correctness Condition

$$\langle L(\preccurlyeq), \alpha, \alpha \circ T_P \circ \rho \rangle$$

$$\alpha \circ T_P = \alpha \circ T_P \circ \rho \circ \alpha$$

Example

```
p(0).
p(1).
p(2) :- p(X), X = 1.
p(3) :- p(X), X = 0.
```

$\langle L(\preccurlyeq), \alpha, \alpha \circ T_P \circ \rho \rangle$

where $\preccurlyeq \; = \; \geq$

$(\alpha \circ T_P)(\{\texttt{p(0)},\texttt{p(1)}\})$

$= [\eta](\{\texttt{p(0)},\texttt{p(1)},\texttt{p(2)},\texttt{p(3)}\})$

$\neq [\eta](\{\texttt{p(2)},\texttt{p(1)},\texttt{p(0)}\})$

$= [\eta](T_P(\{\texttt{p(1)}\}))$

$= (\alpha \circ T_P \circ \rho \circ \alpha)(\{\texttt{p(0)},\texttt{p(1)}\})$

Open Problems

- ★ Current Correctness is much too coarse leading to false negatives

- ★ Verification is hard. Automation is preferable.

Summary

Summary

★ Answer subsumption is a greedy strategy

★ … and Greed is good

★ It's just not always right

★ make sure that $\alpha \circ T_P = T_P^L \circ \alpha$

Please read our paper for more…

- examples,
- technical details,
- related work …

The End

Tabling with Sound Answer Subsumption

Alexander Vandenbroucke
KU Leuven, Belgium
(*e-mail:* alexander.vandenbroucke@kuleuven.be)

Maciej Piróg
KU Leuven, Belgium
(*e-mail:* maciej.pirog@kuleuven.be)

Benoit Desouter
Ghent University, Belgium
(*e-mail:* benoit.desouter@ugent.be)

Tom Schrijvers
KU Leuven, Belgium
(*e-mail:* tom.schrijvers@kuleuven.be)

submitted 29 April 2016; revised 8 July 2016; accepted 22 July 2016

Abstract

Tabling is a powerful resolution mechanism for logic programs that captures their least fixed point semantics more faithfully than plain Prolog. In many tabling applications, we are not interested in the set of all answers to a goal, but only require an aggregation of those answers. Several works have studied efficient techniques, such as lattice-based answer subsumption and mode-directed tabling, to do so for various forms of aggregation.

While much attention has been paid to expressivity and efficient implementation of the different approaches, soundness has not been considered. This paper shows that the different implementations indeed fail to produce least fixed points for some programs. As a remedy, we provide a formal framework that generalises the existing approaches and we establish a soundness criterion that explains for which programs the approach is sound.

KEYWORDS: tabling, answer subsumption, lattice, partial order, mode-directed tabling, denotational semantics, Prolog

1 Introduction

Tabling considerably improves the declarativity and expressiveness of the Prolog language. It removes the sensitivity of SLD resolution to rule and goal ordering, allowing a larger class of programs to terminate. As an added bonus, the memoisation of the tabling mechanism may significantly improve run time performance in exchange for increased memory usage. Tabling has been implemented in a few well-known Prolog systems, such as XSB (Swift and Warren 2010; Swift and Warren 2012),Yap (Santos Costa et al. 2012), Ciao (Chico de Guzmán et al. 2008) and B-Prolog (Zhou 2012), and has been successfully applied in various domains.

Much research effort has been devoted to improving the performance of tabling for


KEEP
CALM
AND
CHECK
BACKUP SLIDES

Existence of the least-fixed point

Theorem [Knaster-Tarski]

Let $\langle L, \leq_L \rangle$ *be a complete lattice, and let* $f : L \rightarrow L$ *be a monotone function, then f has a least fixed point, denoted lfp(f).*

Existence of the least-fixed point

Theorem [Knaster-Tarski]

Let $\langle L, \leq_L \rangle$ *be a complete lattice, and let* $f : L \rightarrow L$ *be a* ***monotone*** *function, then f has a least fixed point, denoted lfp(f).*

$\langle \mathcal{P}(H), \subseteq \rangle$

T_P

for <u>definite</u> programs

Herbrand base = all atoms

Computability of the least fixed point

Fixpoint Theorem [Kleene]

$$lfp(T_P) = \bigcup\{T_P^{0}(\varnothing), T_P^{1}(\varnothing), T_P^{2}(\varnothing), \ldots\}$$

Example

```
p(a). p(b). p(c).
q(X) :- p(X).
```

$I_0 = \varnothing$
$I_1 = T_P(I_0) = \{p(a),p(b),p(c)\}$
$I_2 = T_P(I_1) = \{p(a),p(b),p(c),q(a),q(b),q(c)\}$

Operational independence - Good

tabling mode		semi- lattice
$\alpha = \texttt{min}$	$\longrightarrow$	$\langle U_P, \leq \rangle$
$\alpha = \texttt{max}$	$\longrightarrow$	$\langle U_P, \geq \rangle$
$\alpha = \texttt{all}$	$\longrightarrow$	$\langle P^{fin}(U_P), \subseteq \rangle$
$\alpha = \texttt{lattice}$	$\longrightarrow$	$\langle L, \sqsubseteq \rangle$

Table Lattice

p(a,b,d)

$$L(\preccurlyeq) \triangleq I_P \times U_P^{\,n} \rightarrow U_P^{\,\perp}$$

predicate name	**input arguments**	**aggregated result**

Table Lattice

p(a,b,d)

empty answer

$$L(\preccurlyeq) \triangleq I_P \times U_P{}^n \rightarrow U_P{}^{\perp}$$

predicate name **input arguments** **aggregated result** **$\langle U_P{}^{\perp}, \preccurlyeq \rangle$**

induces lattice

Table Lattice

Embedding function

$$\eta : H \to L(\leqslant)$$

$$\eta(\mathrm{p}(\mathbf{X}, \mathrm{x}))(\mathrm{q}, \mathbf{Y}) = \begin{cases} \mathrm{x} & \textit{if } \mathrm{p} = \mathrm{q} \textit{ and } \mathbf{X} = \mathbf{Y} \\ \perp & \textit{otherwise} \end{cases}$$

Unembedding function

$$\rho : L(\leqslant) \to \mathcal{P}(H)$$

$$\rho(t) = \{\, \mathrm{p}(\mathbf{X}, \mathrm{x}) \mid t(\mathrm{p}, \mathbf{X}) = \mathrm{x} \neq \perp \,\}$$

Extended Immediate Consequence

$$T_P : \mathcal{P}(H) \to \mathcal{P}(H)$$

$$T_P(I) = \bigcup \{ \underbrace{\rho(\bigvee Y)}_{\text{un-embed l.u.b.}} \mid Y \in \underbrace{\mathcal{P}^{fin}(\eta(}_{\text{any derived embedding}} \underbrace{T_P(I)}_{\text{regular immediate consequence}})) \}$$

Table Lattice

Extended Immediate Consequence

$$\check{T}_P : \mathcal{P}(H) \to \mathcal{P}(H)$$

$$\check{T}_P(I) = \bigcup \{ \underbrace{\rho(\bigvee Y)}_{\text{un-embed l.u.b.}} \mid \underbrace{Y \in \mathcal{P}^{fin}(\eta}_{\text{any derived embedding}}(\underbrace{T_P(I)}_{\text{regular immediate consequence}})) \}$$

$\check{T}_P = T_P$ for *linear* *orders*

Table Lattice

Extended Immediate Consequence

$$T_P : \mathcal{P}(H) \longrightarrow \mathcal{P}(H)$$

$$T_P(I) = \bigcup \{ \underbrace{\rho(\bigvee Y)}_{\text{un-embed l.u.b.}} \mid \underbrace{Y \in \mathcal{P}^{fin}(\eta}_{\text{any derived embedding}}(\underbrace{T_P(I)}_{\text{regular immediate consequence}}))\ \}$$

Semantics

$$\boxed{\rho\Big(\bigvee_{x \in lfp(T_P)} \eta(x)\Big)}$$

Extended Immediate Consequence

$$T_P : \mathcal{P}(H) \rightarrow \mathcal{P}(H)$$

$$T_P(I) = \bigcup \{ \underbrace{\rho(\bigvee Y)}_{\text{un-embed l.u.b.}} \mid Y \in \underbrace{\mathcal{P}^{fin}(\eta(}_{\text{any derived embedding}} \underbrace{T_P(I)}_{\text{regular immediate consequence}})) \}$$

Semantics

$$[f](X) \triangleq (\bigvee_{x \in X} f(x))$$

Table Lattice

Extended Immediate Consequence

$$T_P : \mathcal{P}(H) \to \mathcal{P}(H)$$

$$T_P(I) = \bigcup \{ \underbrace{\rho(\bigvee Y)}_{\text{un-embed l.u.b.}} \mid Y \in \underbrace{\mathcal{P}^{fin}(\eta}_{\text{any derived embedding}}(\underbrace{T_P(I)}_{\text{regular immediate consequence}})) \}$$

Semantics

$$\boxed{\rho([\eta](lfp(T_P)))}$$