Logic & Constraint Prog.

Logic program examples and references

IENAC S

Sept. 2015

A small rule-based system for bird identification

// Some rules - Excerpt from a short logic program on amzi.com

(1) $\text{species}(\text{laysan\_albatross}) \leftarrow \text{family}(\text{albatross}) \wedge \text{color}(\text{white}).$

(2) $\text{species}(\text{black\_footed\_albatross}) \leftarrow \text{family}(\text{albatross}) \wedge \text{color}(\text{dark}).$

(3) $\text{species}(\text{whistling\_swan}) \leftarrow \text{family}(\text{swan}) \wedge \text{voice}(\text{muffled\_musical\_whistle}).$

(4) $\text{species}(\text{trumpeter\_swan}) \leftarrow \text{family}(\text{swan}) \wedge \text{voice}(\text{loud\_trumpeting}).$

(5) $\text{species}(\text{canada\_goose}) \leftarrow \text{family}(\text{goose}) \wedge \text{head}(\text{black}) \wedge \text{cheek}(\text{white})$
$\wedge\ ((\text{season}(\text{winter}) \wedge \text{country}(\text{united\_states})) \vee (\text{season}(\text{summer}) \wedge \text{country}(\text{ca}$

...

(6) $\text{family}(\text{albatross}) \leftarrow \text{order}(\text{tubenose}) \wedge \text{size}(\text{large}) \wedge \text{wings}(\text{long\_narrow}).$

(7) $\text{family}(\text{swan}) \leftarrow \text{order}(\text{waterfowl}) \wedge \text{neck}(\text{long}) \wedge \text{color}(\text{white}) \wedge \text{flight}(\text{pondero}$

(8) $\text{family}(\text{goose}) \leftarrow \text{order}(\text{waterfowl}) \wedge \text{size}(\text{plump}) \wedge \text{flight}(\text{powerful}).$

...

(9) $\text{order}(\text{tubenose}) \leftarrow \text{nostrils}(\text{external\_tubular}) \wedge \text{live}(\text{at\_sea}) \wedge \text{bill}(\text{hooked}).$

(10) $\text{order}(\text{waterfowl}) \leftarrow \text{feet}(\text{webbed}) \wedge \text{bill}(\text{flat}).$

+ a few facts about a particular bird:

$(a) : \text{nostrils}(\text{external\_tubular}) \wedge (b) : \text{live}(\text{at\_sea}) \wedge (c) : \text{bill}(\text{hooked}) \wedge (d) : \text{size}(\text{large})$
$\wedge (e) : \text{wings}(\text{long\_narrow}) \wedge (f) : \text{color}(\text{white}).$

A small rule-based system for bird identification

$\Rightarrow$ What can we *deduce* about this bird ?

What if we know instead:

(a') : $\mathsf{head}(\mathsf{black}) \wedge (b')$: $\mathsf{season}(\mathsf{summer}) \wedge (c')$: $\mathsf{cheek}(\mathsf{white})$
$\wedge(d')$: $\mathsf{country}(\mathsf{canada}) \wedge (e')$: $\mathsf{order}(\mathsf{waterfowl}) \wedge (f')$: $\mathsf{size}(\mathsf{plump})$
$\wedge(g')$: $\mathsf{feet}(\mathsf{webbed}) \wedge (h')$: $\mathsf{bill}(\mathsf{flat}) \wedge (i')$: $\mathsf{fligth}(\mathsf{powerful})$.

Note on logical syntax:

- $A \leftarrow B$ is read: "A is true if B is true" (logical implication)
- $B \wedge C$ is true if both B and C are true (logical conjunction)
- $B \vee C$ is true if at least one of B or C are true (logical disjunction)

An excerpt from a movie database

(1) release('*The Blues Brothers*', france, 1980).

(2) director('*John Landis*', '*The Blues Brothers*').

(3) release('*Soul Kitchen*', germany, 2009).

(4) release('*Soul Kitchen*', france, 2010).

(5) director('*Fatih Akin*', '*Soul Kitchen*').

(6) release('*Das Leben der Anderen*', germany, 2006).

(7) release('*Das Leben der Anderen*', france, 2007).

(8) cast('*The Blues Brothers*', '*Dan Aykroyd*', '*“Joliet” Jake Blues*').

(9) cast('*The Blues Brothers*', '*Aretha Franklin*', '*Mrs. Murphy*').

(10) cast('*Soul Kitchen*', '*Adam Bousdoukos*', '*Zinos Kazantsakis*').

(11) cast('*Soul Kitchen*', '*Moritz Bleibtreu*', '*Illias Kazantsakis*').

(12) cast('*Soul Kitchen*', '*Anna Bederke*', '*Lucia Faust*').

(13) cast('*Das Leben der Anderen*', '*Martina Gedeck*', '*Christa-Maria Sielar*

An excerpt from a movie database

(1) release(*'The Blues Brothers'*, france, 1980).
(2) director(*'John Landis'*, *'The Blues Brothers'*).
(3) release(*'Soul Kitchen'*, germany, 2009).
(4) release(*'Soul Kitchen'*, france, 2010).
(5) director(*'Fatih Akin'*, *'Soul Kitchen'*).
(6) release(*'Das Leben der Anderen'*, germany, 2006).
(7) release(*'Das Leben der Anderen'*, france, 2007).
(8) cast(*'The Blues Brothers'*, *'Dan Aykroyd'*, *'"Joliet" Jake Blues'*).
(9) cast(*'The Blues Brothers'*, *'Aretha Franklin'*, *'Mrs. Murphy'*).
(10) cast(*'Soul Kitchen'*, *'Adam Bousdoukos'*, *'Zinos Kazantsakis'*).
(11) cast(*'Soul Kitchen'*, *'Moritz Bleibtreu'*, *'Illias Kazantsakis'*).
(12) cast(*'Soul Kitchen'*, *'Anna Bederke'*, *'Lucia Faust'*).
(13) cast(*'Das Leben der Anderen'*, *'Martina Gedeck'*, *'Christa-Maria Sielar*

Queries A *query* to find the year of the release of Soul Kitchen: release(*'Soul Kitchen'*, C, Y)?
$\Rightarrow$ Prolog's answer: $C =$ germany $\wedge\ Y = 2009 \vee C =$ france $\wedge\ Y = 2010$.

An excerpt from a movie database

$\Rightarrow$ Write queries (and give the answers) to find the following :

1. the director of The Blues Brothers;
2. if Aretha Franklin play in a film by John Landis;
3. actors of movies by Fatih Akin;
4. the directors of movies in which Dan Aykroyd and Anna Bederke were co-stars;
5. if Anna Bederke played in a movie by John Landis or Fatih Akin;
6. actors who are also directors?

   Using $\neq$ and the negation $\neg$:
7. actors who played with Dan Aykroyd;
8. actors who played in more than one movie;
9. directors who where never an actor?
10. actors who never played in a movie directed by Fatih Akin.

An excerpt from a movie database

Rules A *rule* to define a relation $\mathsf{directed}$: $\mathsf{directed}(D, A)$ is true if A played in a movie directed by D:

(16) $\mathsf{directed}(D, A) \leftarrow \mathsf{director}(D, M) \wedge \mathsf{cast}(M, A, R).$

An excerpt from a movie database

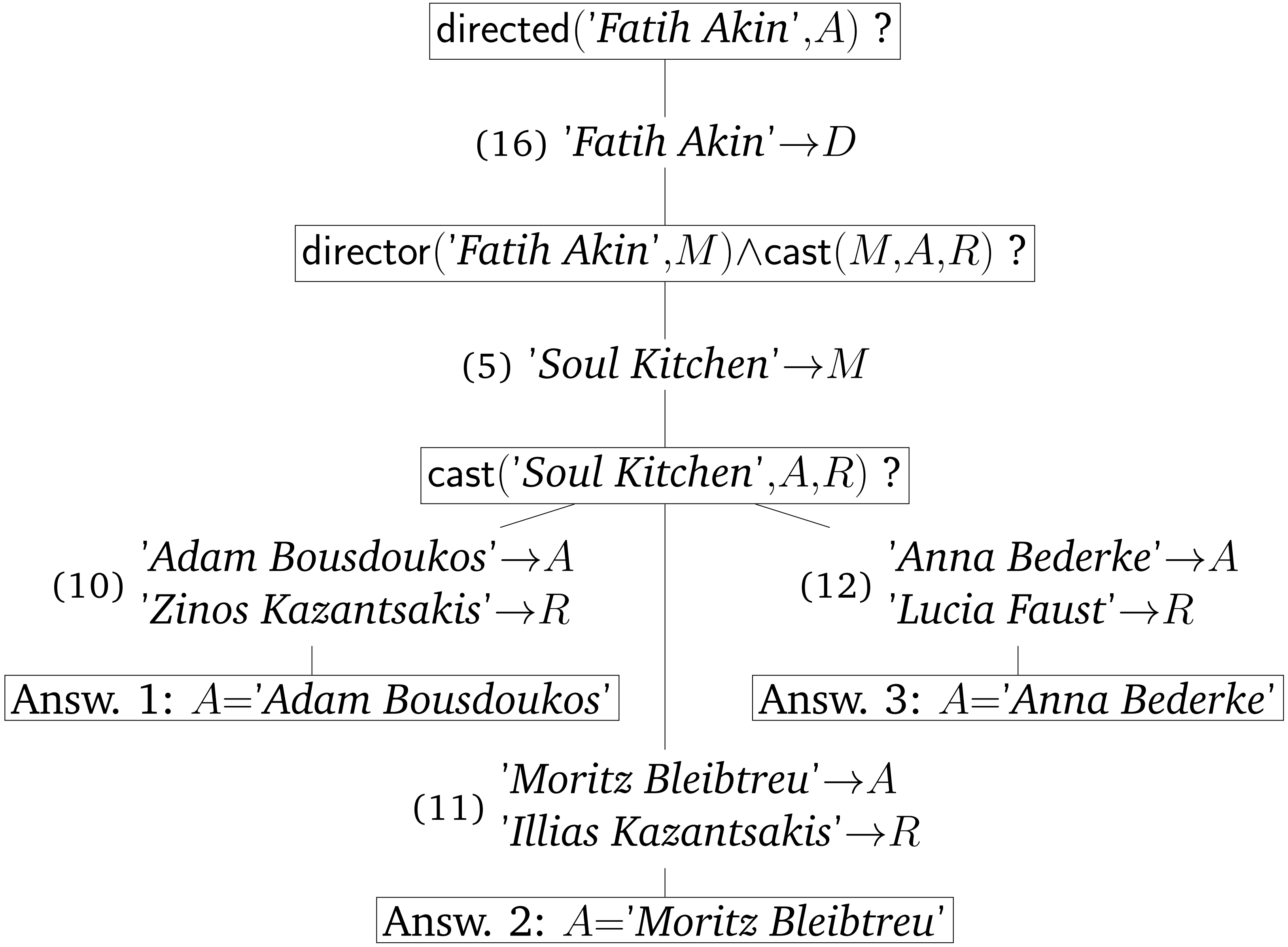
Rules A *rule* to define a relation $\mathsf{directed}$: $\mathsf{directed}(D, A)$ is true if A played in a movie directed by D:

(16) $\mathsf{directed}(D, A) \leftarrow \mathsf{director}(D, M) \wedge \mathsf{cast}(M, A, R).$

$\Rightarrow$ Add rules to the database to define the following:

1. $\mathsf{co\_star}(A1, A2)$ – the actor/actress played in the same movie
2. the films in which play some actor who played in a film by Fatih Akin or John Landis
3. the actors who played in at least two films by Woody Allen
4. movies in which played actors who were never directed by John Landis.

The Prolog search tree


directed('Fatih Akin',A) ?
(16) 'Fatih Akin'→D
director('Fatih Akin',M)∧cast(M,A,R) ?
(5) 'Soul Kitchen'→M
cast('Soul Kitchen',A,R) ?
(10) 'Adam Bousdoukos'→A 'Zinos Kazantsakis'→R
Answ. 1: A='Adam Bousdoukos'
(11) 'Moritz Bleibtreu'→A 'Illias Kazantsakis'→R
Answ. 2: A='Moritz Bleibtreu'
(12) 'Anna Bederke'→A 'Lucia Faust'→R
Answ. 3: A='Anna Bederke'

The Prolog search tree

- Backward chaining ; goals in a conjunction are *proved / executed* from left to right.
- Executing a goal means:
  - replacing it with a condition that makes it true, or
  - simply deleting it if it appears in the database (possibly after some variable instanciation), or
  - returning “false” if it is not possible to make it true.
- Facts / rules of the program are tried in the order in which they appear in the program.
- Prolog systems use a depth-first search strategy.

The Prolog search tree

- Backward chaining ; goals in a conjunction are *proved / executed* from left to right.
- Executing a goal means:
  - replacing it with a condition that makes it true, or
  - simply deleting it if it appears in the database (possibly after some variable instanciation), or
  - returning “false” if it is not possible to make it true.
- Facts / rules of the program are tried in the order in which they appear in the program.
- Prolog systems use a depth-first search strategy.

Question: how many leaves have the trees for the queries

- $\mathsf{cast}(M, A, S) \wedge \mathsf{cast}(M, \text{'}\textit{Anna Bederke}\text{'}, R)$, and
- $\mathsf{cast}(M, \text{'}\textit{Anna Bederke}\text{'}, R) \wedge \mathsf{cast}(M, A, S)$?

A short history of logic programming

1970s: Kowalski (Edinburgh):
the logical formula $\varphi \leftarrow \psi_1 \wedge \ldots \wedge \psi_n$ (φ is true if all the ψ_is are) has a procedural meaning: "In order to prove φ, it is sufficient to prove $\psi_1, \ldots, \psi_n$".
Colmerauer (Aix-Marseille): Prolog,
theorem prover based on the same ideas as Kowalski.

A short history of logic programming

1970s: Kowalski (Edinburgh):
the logical formula $\varphi \leftarrow \psi_1 \wedge \ldots \wedge \psi_n$ (φ is true if all the ψ_is are) has a procedural meaning: “In order to prove φ, it is sufficient to prove $\psi_1, \ldots, \psi_n$”.
Colmerauer (Aix-Marseille): Prolog,
theorem prover based on the same ideas as Kowalski.

end of the 70s: Warren (Edinburgh): Prolog-10,
a fast Prolog implementation; the underlying ideas still are at the core of numerous recent implementations.

A short history of logic programming

begining of 21st century:

- a standard Prolog language (syntax "of Edinburgh");
- far from the ideal of logic programming (a small subset of classical logic);
- **extension to constraints programming**
- numerous implementations, some are open source or free, some have good ide. . . ;
- interface Prolog/other langages (C, java,. . .).
- Prolog widely used, eg.:
  - ∗ Prolog Development Center: airport scheduling (teams, runways, shopfloor,. . .), environmental disaster management,. . .
  - ∗ RDF analysis (Resource Description Framework, W3C)
  - ∗ youbet.com: analysis of information coming from a number of webpages, rules easy to maintain when these pages are modified

A few books

To start with:

Learn Prolog Now! Patrick Blackburn, Johan Bos and Kristina Striegnitz. College Publications, 2006.
On-line version, with lecture slides: `learnprolognow.org`

The Art of Prolog. Leon Sterling and Ehud Shapiro.
MIT Press, 1999 (3rd edition).
⇒ very logical approach (available in French)

Prolog : Programming for Artificial Intelligence. Ivan Bratko.
Addison Wesley, 2001 (3rd edition).
⇒ more computer science oriented (available in French)

More advanced topics:

Programming in Prolog William Clocksin et Christopher Mellish. Spr
1987.

The Craft of Prolog Richard O'Keefe. MIT Press, 1990.