Datalog

P.J. McBrien

Imperial College London

Data is held as extensional predicates

branch		
sortcode	bname	cash
56	'Wimbledon'	94340.45
34	'Goodge St'	8900.67
67	'Strand'	34005.00

account				
no	type	cname	rate?	sortcode
100	'current'	'McBrien, P.'	NULL	67
101	'deposit'	'McBrien, P.'	5.25	67
103	'current'	'Boyd, M.'	NULL	34
107	'current'	'Poulovassilis, A.'	NULL	56
119	'deposit'	'Poulovassilis, A.'	5.50	56
125	'current'	'Bailey, J.'	NULL	56

movement			
mid	no	amount	tdate
1000	100	2300.00	5/1/1999
1001	101	4000.00	5/1/1999
1002	100	-223.45	8/1/1999
1004	107	-100.00	11/1/1999
1005	103	145.50	12/1/1999
1006	100	10.23	15/1/1999
1007	107	345.56	15/1/1999
1008	101	1230.00	15/1/1999
1009	119	5600.00	18/1/1999

branch(56, ‘Wimbledon’, 94340.45).
branch(34, ‘Goodge St’, 8900.67).
branch(67, ‘Strand’, 34005.00).

account(100, ‘current’, ‘McBrien, P.’, null, 67).
account(101, ‘deposit’, ‘McBrien, P.’, 5.25, 67).
account(103, ‘current’, ‘Boyd, M.’, null, 34).
account(107, ‘current’, ‘Poulovassilis, A.’, null, 56).
account(119, ‘deposit’, ‘Poulovassilis, A.’, 5.50, 56).
account(125, ‘current’, ‘Bailey, J.’, null, 56).

movement(1000, 100, 2300.00, 5/1/1999).
movement(1001, 101, 4000.00, 5/1/1999).
movement(1002, 100, −223.45, 8/1/1999).
movement(1004, 107, −100.00, 11/1/1999).
movement(1005, 103, 145.50, 12/1/1999).
movement(1006, 100, 10.23, 15/1/1999).
movement(1007, 107, 345.56, 15/1/1999).
movement(1008, 101, 1230.00, 15/1/1999).
movement(1009, 119, 5600.00, 18/1/1999).

Rules defined as intentional predicates

```
current_account(No, Name, Sortcode) :-
        account(No, 'current', Name, _, Sortcode).
deposit_account(No, Name, Rate, Sortcode) :-
        account(No, 'deposit', Name, Rate, Sortcode).
active_customers(CName, BName) :-
        branch(Sortcode, BName, _),
        account(No, _, CName, _, Sortcode),
        movement(_, No, _, _).
```

Datalog Rules

Datalog rules take the form
Head :- Body.

- Logical semantics: if Body the Head
- Head must be a single predicate
- Body may be any conjunction of predicates.

Naming of predicates and variables

- You cannot use the same name for intentional and extensional predicates
- Convention is the start predicate name with small letter
- Variables start with a capital letter
- A variable that only appears once can be replaced by '_'

Quiz 1: Valid Datalog Knowledgebase

Which Datalog Knowledgebase is invalid?

A

```
single_male(‘Peter’).
married_to(‘Paul’, ‘Jane’).
male(M) :- married_to(M, _).
male(M) :- single_male(M).
female(F) :- married_to(_, F).
female(F) :- single_female(F).
```

B

```
male(‘Peter’).
married_to(‘Paul’, ‘Jane’).
male(M) :- married_to(M, _).
female(F) :- married_to(_, F).
```

C

```
male(‘Peter’).
male(‘Paul’).
female(‘Jane’).
married_to(‘Paul’, ‘Jane’).
```

D

```
married_to(‘Peter’, null).
married_to(‘Paul’, ‘Jane’).
male(M) :- married_to(M, _), isNotNull(M).
female(F) :- married_to(_, F), isNotNull(F).
```

Model-Theoretic Interpretation

```
deposit_account(No, Name, Rate, Sortcode) :-
        account(No, 'deposit', Name, Rate, Sortcode).
account(100, 'current', 'McBrien, P.', null, 67).
account(101, 'deposit', 'McBrien, P.', 5.25, 67).
account(103, 'current', 'Boyd, M.', null, 34).
account(107, 'current', 'Poulovassilis, A.', null, 56).
account(119, 'deposit', 'Poulovassilis, A.', 5.50, 56).
account(125, 'current', 'Bailey, J.', null, 56).
```

Minimal Model

If we can assign any combination of values to the variables, what is the minimum set of predicates that must be true.

Minimal Model

deposit_account(101, 'McBrien, P.', 5.25, 67).

Is not a model, since it implies deposit_account(119, *'Poulovassilis, A.'*, 5.50, 56) *is false, but* deposit_account(119, *'Poulovassilis, A.'*, 5.50, 56) *is true due to the rule for* deposit_account*.*

Model-Theoretic Interpretation

```
deposit_account(No, Name, Rate, Sortcode) :-
        account(No, 'deposit', Name, Rate, Sortcode).
account(100, 'current', 'McBrien, P.', null, 67).
account(101, 'deposit', 'McBrien, P.', 5.25, 67).
account(103, 'current', 'Boyd, M.', null, 34).
account(107, 'current', 'Poulovassilis, A.', null, 56).
account(119, 'deposit', 'Poulovassilis, A.', 5.50, 56).
account(125, 'current', 'Bailey, J.', null, 56).
```

Minimal Model

If we can assign any combination of values to the variables, what is the minimum set of predicates that must be true.

Minimal Model

```
deposit_account(101, 'McBrien, P.', 5.25, 67).
deposit_account(119, 'Poulovassilis, A.', 5.50, 56).
deposit_account(127, 'Poulovassilis, A.', 4.50, 56).
```

Is not a minimal model, since deposit_account(127, *'Poulovassilis, A.'*, 4.50, 56) *could be made false, and the model still be consistent.*

Model-Theoretic Interpretation

```
deposit_account(No, Name, Rate, Sortcode) :-
        account(No, 'deposit', Name, Rate, Sortcode).
account(100, 'current', 'McBrien, P.', null, 67).
account(101, 'deposit', 'McBrien, P.', 5.25, 67).
account(103, 'current', 'Boyd, M.', null, 34).
account(107, 'current', 'Poulovassilis, A.', null, 56).
account(119, 'deposit', 'Poulovassilis, A.', 5.50, 56).
account(125, 'current', 'Bailey, J.', null, 56).
```

Minimal Model

If we can assign any combination of values to the variables, what is the minimum set of predicates that must be true.

Minimal Model

```
deposit_account(101, 'McBrien, P.', 5.25, 67).
deposit_account(119, 'Poulovassilis, A.', 5.50, 56).
```

Is a minimal model

Quiz 2: Datalog Queries

```
active_current_account(No) :-
        account(No, 'current', _, _, _),
        movement(_, No, _, _).
```

What is the minimum model?

A

active_current_account(100).
active_current_account(101).
active_current_account(103).
active_current_account(107).
active_current_account(119).
active_current_account(125).

B

active_current_account(100).
active_current_account(101).
active_current_account(103).
active_current_account(107).
active_current_account(119).

C

active_current_account(100).
active_current_account(103).
active_current_account(107).
active_current_account(125).

D

active_current_account(100).
active_current_account(103).
active_current_account(107).

Datalog$^\neg$: Datalog with Negation

Safe Negation

Use ¬ infront of a predicate to mean that it must not hold.
Any variable that appears in a negated predicate must have previously appeared in a non-negated predicate.

✓Find accounts without any movements

```
dormant_account(No) :-
        account(No, _, _, _, _),
        ¬movement(_, No, _, _).
```

✗Unsafe

```
dormant_account(No) :-
        ¬movement(_, No, _, _).
```

Minimal Model

```
dormant_account(125).
```

Quiz 3: Safe Datalog$^\neg$ Predicates

Which predicate uses safe negation?

A

```
non_current_accounts(No, Type) :-
        account(No, Type, _, _, _),
        ¬Type = 'current'.
```

B

```
non_current_accounts(No, Type) :-
        ¬Type = 'current',
        account(No, Type, _, _, _).
```

C

```
non_current_accounts(No) :-
        ¬Type = 'current',
        account(No, Type, _, _, _).
```

D

```
non_current_accounts(No, Type) :-
        account(No, _, _, _, _),
        ¬Type = 'current'.
```

Quiz 4: Datalog$^\neg$ Queries (1)

```
branch_without_recent_debit(BName) :-
        branch(Sortcode, BName, _),
        account(No, _, _, _, Sortcode),
        ¬account_with_recent_debit(No).
account_with_recent_debit(No) :-
        movement(_, No, Value, TDate),
        Value < 0,
        TDate > 10/1/1999.
```

What is the minimum model?

A

B

branch_without_recent_debit('Wimbledon').

C

branch_without_recent_debit('Goodge St').
branch_without_recent_debit('Strand').

D

branch_without_recent_debit('Wimbledon').
branch_without_recent_debit('Goodge St').
branch_without_recent_debit('Strand').

Quiz 5: Datalog¬ Queries (2)

```
branch_without_recent_debit(BName) :-
        branch(Sortcode, BName, _),
        ¬branch_with_recent_debit(Sortcode).
branch_with_recent_debit(Sortcode) :-
        account(No, _, _, _, Sortcode),
        movement(_, No, Value, TDate),
        Value < 0,
        TDate > 10/1/1999.
```

What is the minimum model?

A

B

branch_without_recent_debit('Wimbledon').

C

branch_without_recent_debit('Goodge St').
branch_without_recent_debit('Strand').

D

branch_without_recent_debit('Wimbledon').
branch_without_recent_debit('Goodge St').
branch_without_recent_debit('Strand').

Projection

π

RA projection is performed by only using a subset of rule body variables in the head of a rule.

π_{sortcode} account

```
account_sortcode(Sortcode) :-
        account(_, _, _, _, Sortcode).
```

Minimal Model

```
account_sortcode(34).
account_sortcode(56).
account_sortcode(67).
```

Selection

σ

RA selection is performed by naming a variable more than once, or by putting a data value in the rule body.

$\sigma_{\mathsf{amount}>1000}$ movement

```
big_credit(Mid, No, Amount, Date) :-
      movement(Mid, No, Amount, Date),
      Amount > 1000.
```

Minimal Model

```
big_credit(1000, 100, 2300.00, 5/1/1999).
big_credit(1001, 101, 4000.00, 5/1/1999).
big_credit(1008, 101, 1230.00, 15/1/1999).
big_credit(1009, 119, 5600.00, 18/1/1999).
```

Product

×

RA product is performed by naming two predicates in the rule body.

branch $\times$ $\sigma_{\text{rate}>0}$account

```
product_example(BSortcode, BName, Cash, No, Type, CName, Rate, ASortcode) :-
        branch(BSortcode, BName, Cash),
        account(No, Type, CName, Rate, ASortcode),
        Rate > 0.
```

Minimal Model

(56, ‘Wimbledon’, 94340.45, 101, ‘deposit’, ‘McBrien, P.’, 5.25, 67)
(56, ‘Wimbledon’, 94340.45, 119, ‘deposit’, ‘Poulovassilis, A.’, 5.50, 56)
(34, ‘Goodge St’, 8900.67, 101, ‘deposit’, ‘McBrien, P.’, 5.25, 67)
(34, ‘Goodge St’, 8900.67, 119, ‘deposit’, ‘Poulovassilis, A.’, 5.50, 56)
(67, ‘Strand’, 34005.00, 101, ‘deposit’, ‘McBrien, P.’, 5.25, 67)
(67, ‘Strand’, 34005.00, 119, ‘deposit’, ‘Poulovassilis, A.’, 5.50, 56)

Join

$\bowtie$

RA join is performed by naming two predicates in the rule body, and then comparing their attributes.

$\pi_{\text{bname,cname}}\,\sigma_{\text{branch.sortcode=account.sortcode}}(\text{branch} \times \text{account})$

```
branch_customers(BName, CName) :-
        branch(BSortcode, BName, _),
        account(_, _, CName, _, ASortcode),
        BSortcode = ASortcode.
```

$\equiv$

```
branch_customers(BName, CName) :-
        branch(Sortcode, BName, _),
        account(_, _, CName, _, Sortcode).
```

Minimal Model

```
branch_customers('Wimbledon', 'Poulovassilis, A.').
branch_customers('Wimbledon', 'Bailey, J.').
branch_customers('Goodge St', 'Boyd, M.').
branch_customers('Strand', 'McBrien, P.').
```

Quiz 6: Translating RA to Datalog

$\pi_{\text{bname}}\, \sigma_{\text{account.sortcode}=\text{branch.sortcode} \wedge \text{type}=\text{'deposit'}} (\text{account} \times \text{branch})$

Which datalog rule for **query** is *not* equivalent to the above RA query?

A

```
query(BName) :-
        account(_, 'deposit', _, _, Sortcode),
        branch(Sortcode, BName, _).
```

B

```
query(BName) :-
        branch(Sortcode1, BName, _),
        account(_, 'deposit', _, _, Sortcode2),
        Sortcode1 = Sortcode2.
```

C

```
query(BName) :-
        branch(_, BName, _).
query(BName) :-
        branch(Sortcode, BName, _),
        account(_, 'deposit', _, _, Sortcode).
```

D

```
query(BName) :-
        branch(Sortcode, BName, _),
        deposit_branch(Sortcode).
deposit_branch(Sortcode) :-
        account(_, 'deposit', _, _, Sortcode).
```

Quiz 7: Self Joins

```
query(CName, CAcc, DAcc) :-
        account(DAcc, ‘deposit’, CName, _, _),
        account(CAcc, ‘current’, CName, _, _).
```

account				
no	type	cname	rate?	sortcode
100	'current'	'McBrien, P.'	NULL	67
101	'deposit'	'McBrien, P.'	5.25	67
103	'current'	'Boyd, M.'	NULL	34
107	'current'	'Poulovassilis, A.'	NULL	56
119	'deposit'	'Poulovassilis, A.'	5.50	56
125	'current'	'Bailey, J.'	NULL	56

What is the result of the Datalog query?

A

CName	CAcc	DAcc

B

CName	CAcc	DAcc
‘McBrien, P.’	100	101
‘Poulovassilis, A.’	107	119

C

CName	CAcc	DAcc
‘McBrien, P.’	101	100
‘Poulovassilis, A.’	119	107

D

CName	CAcc	DAcc
‘McBrien, P.’	100	101
‘Boyd, M.’	103	null
‘Poulovassilis, A.’	107	119
‘Bailey, J.’	103	null

Union

$\cup$

RA union is performed by having more than one rule definition for an intentional predicate.

$\sigma_{\text{amount}>1000}$ movement $\cup$ $\sigma_{\text{amount}<-100}$ movement

```
big_movement(Mid, No, Amount, Date) :-
        movement(Mid, No, Amount, Date),
        Amount > 1000.
big_movement(Mid, No, Amount, Date) :-
        movement(Mid, No, Amount, Date),
        Amount < −100.
```

Minimal Model

```
big_movement(1000, 100, 2300.00, 5/1/1999).
big_movement(1001, 101, 4000.00, 5/1/1999).
big_movement(1002, 100, −223.45, 8/1/1999).
big_movement(1008, 101, 1230.00, 15/1/1999).
big_movement(1009, 119, 5600.00, 18/1/1999).
```

Difference

$-$

RA difference is performed using a negation on the predicate being ‘subtracted’: need Datalog $\neg$.

π_{no} account $-$ π_{no} movement

```
dormant_account(No) :-
        account(No, _, _, _, _),
        ¬movement(_, No, _, _).
```

Minimal Model

dormant_account(125).

Worksheet: Datalog

branch		
sortcode	bname	cash
56	'Wimbledon'	94340.45
34	'Goodge St'	8900.67
67	'Strand'	34005.00

movement			
mid	no	amount	tdate
1000	100	2300.00	5/1/1999
1001	101	4000.00	5/1/1999
1002	100	-223.45	8/1/1999
1004	107	-100.00	11/1/1999
1005	103	145.50	12/1/1999
1006	100	10.23	15/1/1999
1007	107	345.56	15/1/1999
1008	101	1230.00	15/1/1999
1009	119	5600.00	18/1/1999

account				
no	type	cname	rate?	sortcode
100	'current'	'McBrien, P.'	NULL	67
101	'deposit'	'McBrien, P.'	5.25	67
103	'current'	'Boyd, M.'	NULL	34
107	'current'	'Poulovassilis, A.'	NULL	56
119	'deposit'	'Poulovassilis, A.'	5.50	56
125	'current'	'Bailey, J.'	NULL	56

key branch(sortcode)
key branch(bname)
key movement(mid)
key account(no)

movement(no) $\overset{fk}{\Rightarrow}$ account(no)

account(sortcode) $\overset{fk}{\Rightarrow}$ branch(sortcode)