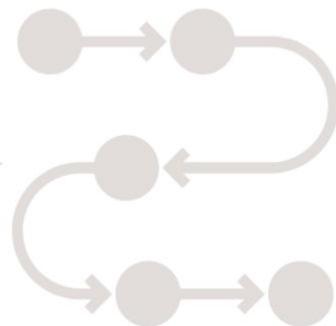


```
if
  item == vehicle
then
  may_park = False
```

Natural language generation and processing for the legal domain

Inari Listenmaa, Martin Strecker, Warrick Macmillan



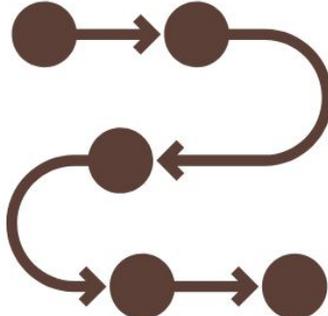
§1
*Thou shalt not
park a vehicle*

Computational law

```
if  
  item == vehicle  
then  
  may_park = False
```



FORMAL VERIFICATION



END-USER APPLICATIONS



NATURAL LANGUAGE

§1
Thou shalt not park a vehicle

L4: DSL for legal domain

Programming language interface

```
1 rule <s64_3_1a>
2 for dt: Data, db: DataBreach, indiv: Individual
3 if dataOfBreach dt db
4 && ( isNameOf indiv dt
5     || isAliasOf indiv dt
6     || isIdentificationNumberOf indiv dt
7     || dataSchedulePart1 indiv dt && not dataSchedulePart2 indiv dt
8 )
9 then significantHarm db indiv
10
11 rule <s64_3_1b>
12 for dt: Data, db: DataBreach, indiv: Individual, n: Integer
13 if dataOfBreach dt db
14 && ( isAccountIdentifier indiv dt
15     || isAccessCode indiv dt
16 )
17 then significantHarm db indiv
18
19 decl numberOfAffectedIndividuals: DataBreach -> Integer
20
21 rule <s64_4>
22 for db: DataBreach
23 if numberOfAffectedIndividuals db >= 500
24 then exceedsPrescrNumberOfIndividuals db
25
26
27 U:*** pdpa.l4 Top L25
Beginning of buffer
```

Natural language interface

§ NDB Qualification

(Act §26B.1) IT IS a Notifiable Data Breach

AKA "NDB"

(Act §26A) IF a data breach occurred

IS any unauthorised

OR access

OR use

OR disclosure

OR copying

OR modification

OR disposal of personal data

OR loss of storage medium on which personal data is stored in circumstances where

(Act §26B.4) UNLESS the data breach occurred only within an organisation

(Act §26B.1.a) AND it results in, or is likely to result in, significant harm to an affected individual

(Act §26B.2.a) IF the data breach is in relation to any prescribed personal data or class of

(Regs §3.1) IF the data breach relates to

(Regs §3.1.a) the individual's

(Regs §3.1.a) full name

(Regs §3.1.a) alias

(Regs §3.1.a) OR identification

(Regs §3.1.a) AND any of the prescribed personal data or class of

(Regs §3.1.a) set out in Part 1 of the Schedule

(Regs §3.1.a) subject to Part 2 of the Schedule

L4: DSL for legal domain

Programming language interface

```
pdpa.l4
1 rule <s64_3_1a>
2 for dt: Data, db: DataBreach, indiv: Individual
3 if dataOfBreach dt db
4 && ( isNameOf indiv dt
5     || isAliasOf indiv dt
6     || isIdentificationNumberOf indiv dt
7     || dataSchedulePart1 indiv dt && not dataSchedulePart2 indiv dt
8     )
9 then significantHarm dt db
10
11 rule <s64_3_1b>
12 for dt: Data, db: DataBreach, indiv: Individual, n: Integer
13 if dataOfBreach dt db
14 && ( isAccountIdentifier indiv dt
15     || isAccessCode indiv dt
16     )
17 then significantHarm db indiv
18
19 decl numberOfAffectedIndividuals: DataBreach -> Integer
20
21 rule <s64_4>
22 for db: DataBreach
23 if numberOfAffectedIndividuals db >= 500
24 then exceedsPrescrNumberOfIndividuals db
25
26
27
U:+++ pdpa.l4 Top L25 (Haskell +3 company EIDoc Wrap)
Beginning of buffer
```

CORE L4

Natural language interface

§ NDB Qualification

(Act §26B.1) IT IS a Notifiable Data Breach

AKA "NDB"

(Act §26A) IF a data breach occurred

IS any unauthorised

OR access

OR use

OR disclosure

OR copying

OR modification

NATURAL L4

(Act §26B.4) UNLESS the data breach occurred only within an organisation

(Act §26B.1.a) AND it results in, or is likely to result in, significant harm to an affected individual

(Act §26B.2.a) IF the data breach is in relation to any prescribed personal data or class of personal data

(Regs §3.1) IF the data breach relates to

(Regs §3.1.a) the individual's

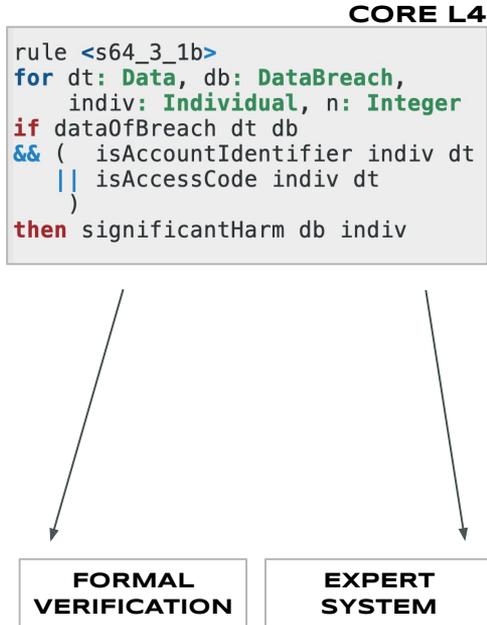
(Regs §3.1.a) alias

(Regs §3.1.a) identification

(Regs §3.1.a) AND any of the prescribed personal data or class of personal data set out in Part 1 of the Schedule

(Regs §3.1.a) subject to Part 2 of the Schedule

Current system



NATURAL L4

| | | |
|-----|-------------------------------------|---|
| IF | <input type="checkbox"/> | the data breach relates to |
| | <input checked="" type="checkbox"/> | the individual's |
| | | , |
| | <input checked="" type="checkbox"/> | full name |
| | | <input type="checkbox"/> alias |
| | | OR <input type="checkbox"/> identification number |
| AND | <input type="checkbox"/> | any of the prescribed personal data or classes of personal data |

A dashed arrow points downwards from the bottom center of the NATURAL L4 form to a box labeled 'NATURAL LANGUAGE'.

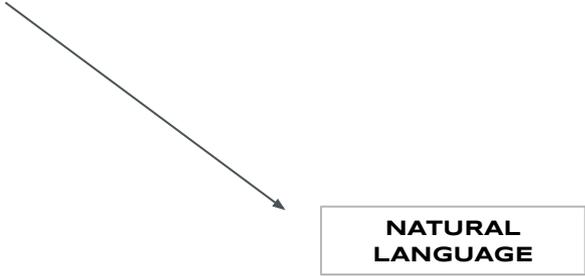
**NATURAL
LANGUAGE**

Previous work: NLG from Core L4

CORE L4

```
rule <s64_3_1b>
for dt: Data, db: DataBreach,
  indiv: Individual, n: Integer
if dataOfBreach dt db
&& ( isAccountIdentifier indiv dt
  || isAccessCode indiv dt
)
then significantHarm db indiv

lexicon
significantHarm @
"{db} causes significant harm to {indiv}"
```



Input:

- Logical predicates
- CNL descriptions

Output:

- Natural language

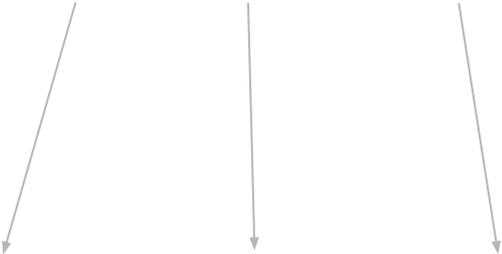
Current work: Logic from Natural L4

CORE L4

```
rule <s64_3_1b>
for dt: Data, db: DataBreach,
  indiv: Individual, n: Integer
if dataOfBreach dt db
&& ( isAccountIdentifier indiv dt
  || isAccessCode indiv dt
)
then significantHarm db indiv
```

NATURAL L4

| | | |
|-----|-------------------------------------|---|
| IF | <input type="checkbox"/> | the data breach relates to |
| | <input checked="" type="checkbox"/> | the individual's |
| | | , <input checked="" type="checkbox"/> full name |
| | | OR <input type="checkbox"/> alias |
| | | <input type="checkbox"/> identification number |
| AND | <input type="checkbox"/> | any of the prescribed personal data or classes of personal data |



APPLICATIONS

Current work: Logic from Natural L4

CORE L4

```
rule <s64_3_1b>
for dt: Data, db: DataBreach,
  indiv: Individual, n: Integer
if dataOfBreach dt db
&& ( isAccountIdentifier indiv dt
  || isAccessCode indiv dt
)
then significantHarm db indiv
```

NATURAL L4

| | | |
|-----|-------------------------------------|---|
| IF | <input type="checkbox"/> | the data breach relates to |
| | <input checked="" type="checkbox"/> | the individual's |
| | | , <input checked="" type="checkbox"/> full name |
| | | <input type="checkbox"/> alias |
| | | OR <input type="checkbox"/> identification number |
| AND | <input type="checkbox"/> | any of the prescribed personal data or classes of personal data |

APPLICATIONS

Current work: Logic from Natural L4

CORE L4

```
rule <s64_3_1b>
for dt: Data, db: DataBreach,
  indiv: Individual, n: Integer
if dataOfBreach dt db
&& ( isAccountIdentifier indiv dt
  || isAccessCode indiv dt
)
then significantHarm db indiv
```

NATURAL L4

| | | |
|-----|-------------------------------------|---|
| IF | <input type="checkbox"/> | the data breach relates to |
| | <input checked="" type="checkbox"/> | the individual's |
| | | , |
| | | OR |
| AND | <input type="checkbox"/> | any of the prescribed personal data or classes of personal data |
| | <input checked="" type="checkbox"/> | full name |
| | <input type="checkbox"/> | alias |
| | <input type="checkbox"/> | identification number |

APPLICATIONS

Current work: Logic from Natural L4

NATURAL L4

| | | | | | | |
|-----|--------------------------|---|-------------------------------------|------------------|-------------------------------------|-----------------------|
| IF | <input type="checkbox"/> | the data breach relates to | <input checked="" type="checkbox"/> | the individual's | <input checked="" type="checkbox"/> | full name |
| | | | | , | <input type="checkbox"/> | alias |
| | | | | OR | <input type="checkbox"/> | identification number |
| AND | <input type="checkbox"/> | any of the prescribed personal data or classes of personal data | | | | |

LOGIC

Current work: Logic from Natural L4

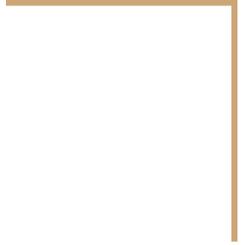
NATURAL L4

| | | | | | | |
|-----|--------------------------|---|-------------------------------------|------------------|-------------------------------------|-----------------------|
| IF | <input type="checkbox"/> | the data breach relates to | <input checked="" type="checkbox"/> | the individual's | <input checked="" type="checkbox"/> | full name |
| | | | | , | <input type="checkbox"/> | alias |
| | | | | OR | <input type="checkbox"/> | identification number |
| AND | <input type="checkbox"/> | any of the prescribed personal data or classes of personal data | | | | |

Research question: is Natural L4 a good format for NLG and logic?

LOGIC

Logic from Natural L4



Personal Data Protection Act — Data Breach Notification Obligation

This is a technology prototype, not legal advice.

This prototype is intended to explore the feasibility of a spreadsheet-as-interface for Rules As Code.

Sources: [Act Part VIA Regulations](#) [Advisory Guidelines part 20 Guide](#)

[Unit Tests](#)

PHASE 1 ASSESS



"My organisation may have experienced a data breach.
What am I supposed to do now?"

| I Assessment | |
|---|--|
| (Act §46C.2) (Act §41.4) (Act §46C.2) (Act §46C.2) (Act §46C.2) (Act §46C.1) | <p>Organisation ("you") is not a Public Agency</p> <p>UPON MUST become aware a data breach may have occurred WITHIN 30 days</p> <p>IF the breach is a Notifiable Data Breach (NDB) as defined in section 4 of PDPA(A) 2020 §13</p> <p>HENCE LEST the PDPC MAY NOTIFY you with a demand for an explanation of your inaction</p> <p>HENCE you MUST NOTIFY the PDPC with an explanation of your inaction</p> |
| (Act §46C.3) (Act §46C.3) (Act §46C.3) (Act §46C.3) (Act §46C.3) | <p>Organisation ("you") is not processing personal data on behalf of and for the purposes of a public agency</p> <p>MUST without undue delay become aware a data breach involving a client organisation may have occurred</p> <p>WHEN the data breach occurs on or after the date of commencement of PDPA(A) 2020 §13</p> <p>NOTIFY the Organisation for which you act as a Data Intermediary</p> |
| II Notification | |
| (Act §46B.1) | <p>IS a data breach IS a Notifiable Data Breach</p> |
| (Act §46A) | <p>IF a data breach occurred</p> <p>IS any unauthorised</p> <p>OR access</p> <p>OR use</p> <p>OR disclosure</p> <p>OR copying</p> <p>OR modification</p> <p>OR disposal of personal data</p> <p>OR loss of storage medium on which personal data is stored in circumstances where the unauthorised</p> <p>OR access</p> <p>OR use</p> <p>OR disclosure</p> <p>OR copying</p> <p>OR modification</p> <p>OR disposal of the personal data is likely to occur</p> |
| (Act §46B.4) | <p>UNLESS the data breach occurred only within an organisation</p> |
| (Act §46B.1a) (Act §46B.1a) (Regs §3.1) (Act §3.1.1) (Act §3.1.1) (Act §3.1.1) (Act §3.1.1) (Act §3.1.1) (Act §3.1.1) | <p>AND it results in, or is likely to result in, significant harm to an affected individual</p> <p>IF the data breach is in relation to any prescribed personal data or class of personal data relating to the individual</p> <p>IF the data breach relates to</p> <p>the individual's full name</p> <p>OR alias</p> <p>OR identification number</p> <p>AND any of the prescribed personal data or classes of personal data relating to the individual set out in Part 1 of the schedule</p> |
| (Act §46B.1b) (Regs §3.1.1b) (Act §3.1.1b) (Act §3.1.1b) (Act §3.1.1b) (Act §3.1.1b) | <p>OR in other prescribed circumstances</p> <p>MEANS all of the following personal data relating to an individual's account with an organisation:</p> <p>the individual's account identifier</p> <p>INCLUDES an account number</p> <p>OR a number assigned to any account the individual has with an organisation that is a bank or finance company.</p> <p>AND any</p> <p>security code</p> <p>access code</p> <p>response to a security question</p> <p>biometric data</p> <p>OR other data that is used to allow or required to access to the individual's account.</p> |
| (Act §46B.1b) (Act §46B.1a) (Act §46B.1b) | <p>OR it is, or is likely to be, of a significant scale</p> <p>IF the data breach affects not fewer than the prescribed number of affected individuals</p> <p>OR in other prescribed circumstances.</p> <p>500 (Regs §3.1)</p> |

| PHASE 2 NOTIFY | |
|--|--|
| "Do I have to notify the PDPC?" | |
| <p>Yes <input type="checkbox"/> No <input type="checkbox"/> Yes <input type="checkbox"/> If it is a Notifiable Data Breach</p> <p>No <input type="checkbox"/> otherwise</p> | |
| III Notification | |
| (Act §46D.0) (Act §41.4) (Act §46D.0) (Act §46D.0) (Act §46D.0) (Act §46D.0) (Act §46D.0) | <p>IF you are a Public Agency</p> <p>IF it is an NDB</p> <p>UNLESS you are a Public Agency</p> <p>NOTIFY the PDPC in the form and manner specified at www.pdpc.gov.sg with a Notification Message and (optional) a list of individuals for whom notification waiver is sought together with justification for why</p> <p>HENCE the PDPC MAY NOTIFY you with a list of individuals to exclude from notification AKA the PDPC Exclusion List</p> |
| (Act §46D.5a) (Act §46D.5a) (Act §46D.5a) (Act §46D.5b) (Act §46D.6a) | <p>GIVEN an individual WHO is affected by an NDB</p> <p>UNLESS it is unlikely that the notifiable data breach will result in significant harm to the affected individual</p> <p>AKA "Unlikely"</p> <p>IF the organisation has taken any action</p> <p>OR the organisation already implemented any technological measure</p> <p>to render it unlikely that the notifiable data breach will result in significant harm to the individual</p> |
| (Act §46D.5a) (Act §46D.5a) (Act §46D.5b) (Act §46D.6a) | <p>TO DO Where the data breach involves information related to admission matters or the identification of vulnerable individuals, organisations should first notify the Commission for guidance on notifying affected individuals.</p> |
| (Act §46D.0) (Act §41.4) (Act §46D.0) | <p>IV to Notifiable Individuals</p> <p>IF it is an NDB</p> <p>UNLESS you are a Public Agency</p> <p>You MUST WITHIN 3 days</p> <p>NOTIFY each of the Notifiable Individuals in any manner that is reasonable in the circumstances with a message obeying a certain format</p> <p>WHICH IS the set of individuals affected by the NDB</p> <p>LESS the individuals who are deemed unlikely</p> <p>LESS the individuals on the PDPC Exclusion List</p> <p>AND provide to the PDPC an explanation for why your notification was late</p> |
| Part 1 of the Schedule | |
| <p>any of the prescribed personal data or classes of personal data relating to the individual</p> <p>1 The amount of any wages, salary, fee, commission, bonus, gratuity, allowance or other remuneration paid or payable to the individual by any person, whether under a contract of service or not</p> <p>2 The income of the individual from the sale of any goods or property.</p> <p>3 The number of any credit card, charge card or debit card issued to or in the name of the individual.</p> <p>4 The number assigned to any account the individual has with any organisation that is a bank or finance company.</p> <p>5 Any information that identifies, or is likely to lead to the identification of, the individual as a child or young person who —</p> <p>5.a is or had been the subject of any investigation under the CPA;</p> <p>5.b is or had been arrested, on or after 1 July 2020, for an offence committed under any written law;</p> <p>5.c is or had been taken into care or custody by the Director-General of Social Welfare, a protective, any officer generally or specially authorised in that behalf in writing by the Director-General or is attending or had attended a family programme in relation to an application to be made under section 50 of the CPA;</p> <p>5.d is or was the subject of an order made by a court under the CPA; or</p> <p>5.e is or has been concerned in any proceedings in any court or an appeal from any court, whether the individual is the person against or in respect of whom the proceedings are taken or a witness</p> <p>6 Any information that identifies, or is likely to lead to the identification of —</p> <p>6.a the individual who has been or is the subject of any investigation, examination, assessment or treatment under the VAA relating to whether the individual is a vulnerable adult experiencing or the individual as a vulnerable adult who has been committed to a place of temporary care and protection or place of safety or to the care of a fit person under the VAA;</p> <p>6.b the individual as a vulnerable adult who is the subject of an order made by a court under the VAA;</p> <p>6.c a place of temporary care and protection or place of safety in which an individual or a vulnerable adult mentioned in sub-paragraph (a), (b) or (c) is committed, or the location of such a place of fit person under whose care an individual or a vulnerable adult mentioned in sub-paragraph (a), (b) or (c) is placed, or the location of the premises of such a fit person.</p> <p>7 Any private key or of relating to the individual that is used or may be used —</p> <p>7.a to create a secure electronic record or secure electronic signature;</p> <p>7.b to verify the integrity of a secure electronic record; or</p> <p>7.c to verify the authenticity or integrity of a secure electronic signature.</p> <p>8 The net worth of the individual.</p> <p>9 The deposit of moneys by the individual with any organisation.</p> <p>10 The withdrawal by the individual of moneys deposited with any organisation.</p> <p>11 The granting by an organisation of advances, loans and other facilities by which the individual, being a customer of the organisation, has access to funds or financial guarantees.</p> | |

Personal Data Protection Act — Data Breach Notification Obligation

This is a technology prototype, not legal advice.

This prototype is intended to explore the feasibility of a spreadsheet-as-interface for Rules As Code.

Sources: [Act Part VIA Regulations](#) [Advisory Guidelines part 20 Guide](#) [Unit Tests](#)

PHASE 1 ASSESS

"My organisation may have experienced a data breach. What am I supposed to do now?"

Assessment

Organisation ("you") is not a Public Agency becoming aware a data breach may have occurred

UPON MUST WITHIN 30 days

GOTO G94

IF **Yes** IF it is a Notifiable Data Breach (NDB) (Reg 5.1)

HENCE **GOTO G142**

NOTIFY you with a demand for an explanation of your reaction

HENCE you MUST NOTIFY the PDPC with an explanation of your reaction

Data Intermediary ("you") is not processing personal data on behalf of and for the purposes of a public agency without undue delay

UPON becoming aware a data breach involving a client Organisation may have occurred

WHEN the data breach occurs on or after the date of commencement of PDPA(A) 2020 §11

NOTIFY the Organisation for which you act as a Data Intermediary

Notification

IF **Yes** IF it is a Notifiable Data Breach (NDB)

RETURN RESULT

IF a data breach occurred

IS any unauthorised

OR access

OR use

OR disclosure

OR copying

OR modification

OR disposal

of personal data

OR loss of storage medium on which personal data is stored in circumstances where the unauthorised

OR access

OR use

OR disclosure

OR copying

OR modification

OR disposal

of the personal data is likely to occur

UNLESS the data breach occurred only within an organisation

AND it results in, or is likely to result in, significant harm to an affected individual

IF the data breach is in relation to any prescribed personal data or class of personal data relating to the individual

IF the data breach relates to

the individual's full name

alias

OR identification number

AND any of the prescribed personal data or classes of personal data relating to the individual set out in Part 1 of the schedule

OR in other prescribed circumstances

MEANS all of the following personal data relating to an individual's account with an organisation:

OR the individual's account identifier

INCLUDES an account number

OR a number assigned to any account the individual has with an organisation that is a bank or finance company.

AND any

security code

access code

response to a security question

biometric data

OR other data

that is used to allow or required to use of the individual's account.

OR it is, or is likely to be, of a significant scale

IF if the data breach affects not fewer than **the prescribed number of affected individuals**

OR in other prescribed circumstances.

500 (Regs 5.3)

PHASE 2 NOTIFY

"Do I have to notify the PDPC?"

IF **Yes** IF it is a Notifiable Data Breach

IF **No** otherwise

Notification

IF **Yes** IF it is a Notifiable Data Breach (NDB)

UNLESS you are a Public Agency

NOTIFY the PDPC in the form and manner specified at www.pdpc.gov.sg with a Notification Message (in the case of a Notifiable Data Breach) and (optional) a list of individuals for whom notification waiver is sought together with justification for why

HENCE the PDPC MAY NOTIFY you with a list of individuals to exclude from notification **AKA the PDPC Exclusion List**

AND

"Do I have to notify the individuals involved?"

GIVEN an individual WHO is affected by an NDB

UNLESS it is unlikely that the notifiable data breach will result in significant harm to the affected individual

AKA "Chuckle"

IF the organisation has taken any action

OR the organisation already implemented any technological measure

to render it unlikely that the notifiable data breach will result in significant harm to the individual

(Adv p28)

TODO Where the data breach involves information related to admission matters or the identification of vulnerable individuals, organisations should first notify the Commission for guidance on notifying affected individuals.

IF it is an NDB

UNLESS you are a Public Agency

You MUST WITHIN 3 days

NOTIFY each of the Notifiable Individuals in any manner that is reasonable in the circumstances with a message obeying a certain format

WHICH the set of individuals affected by the NDB

LESS the individuals who are deemed unlikely

OR LESS the individuals on the PDPC Exclusion List

AND provide to the PDPC an explanation for why your notification was late

Part 1 of the Schedule

any of the prescribed personal data or classes of personal data relating to the individual

1 The amount of any wages, salary, fee, commission, bonus, gratuity or other remuneration paid or payable to the individual by any person, whether under a contract of service or not

2 The income of the individual from the sale of any goods or property.

3 The number of any credit card, charge card or debit card issued to or in the name of the individual.

4 The number assigned to any account the individual has with any organisation that is a bank or finance company.

5 Any information that identifies, or is likely to lead to the identification of, the individual as a child or young person who —

5.a is or had been the subject of any investigation under the CPA;

5.b is or had been arrested, on or after 1 July 2020, for an offence committed under any written law;

5.c is or had been taken into care or custody by the Director-General of Social Welfare, a protective, any officer generally or specially authorised in that behalf in writing by the Director-General or is attending or had attended a family programme in relation to an application to be made under section 50 of the CPA;

5.d is or was the subject of an order made by a court under the CPA; or

5.e is or had been concerned in any proceedings in any court or an appeal from any court, whether the individual is the person against or in respect of whom the proceedings are taken or a witness

6 Any information that identifies, or is likely to lead to the identification of —

6.a the individual who has been or is the subject of any investigation, examination, assessment or treatment under the VAA relating to whether the individual is a vulnerable adult experiencing or the individual as a vulnerable adult who has been committed to a place of temporary care and protection or place of safety or to the care of a fit person under the VAA;

6.b the individual as a vulnerable adult who is the subject of an order made by a court under the VAA;

6.c a place of temporary care and protection or place of safety in which an individual or a vulnerable adult mentioned in sub-paragraph (a), (b) or (c) is committed, or the location of such a place of fit person under whose care an individual or a vulnerable adult mentioned in sub-paragraph (a), (b) or (c) is placed, or the location of the premises of such a fit person.

7 Any private key or of relating to the individual that is used or may be used —

7.a to create a secure electronic record or secure electronic signature;

7.b to verify the integrity of a secure electronic record; or

7.c to verify the authenticity or integrity of a secure electronic signature.

8 The net worth of the individual.

9 The deposit of moneys by the individual with any organisation.

10 The withdrawal by the individual of moneys deposited with any organisation.

11 The granting by an organisation of advances, loans and other facilities by which the individual, being a customer of the organisation, has access to funds or financial guarantees.

Natural L4: Single rule

| § Assessment | |
|--------------|---|
| (Act §26C.2) | EVERY Organisation |
| (Act §4.1.c) | WHO is not a Public Agency |
| (Act §26C.2) | MUST WITHIN 30 days |
| (Act §26C.2) | UPON becoming aware a data breach may have occurred |
| (Act §26C.1) | IF the data breach occurs after the date of commencement of PDP(A)A 2020 §13 |
| (Act §26C.2) | → assess if it is a Notifiable Data Breach |
| | HENCE <i>Notification</i> |

KEYWORDS

FREE TEXT

Natural L4 to logic

| | | | |
|--------------|-------|---|-----------------------------------|
| (Act §26C.2) | EVERY | Organisation | |
| (Act §4.1.c) | WHO | is not | a Public Agency |
| (Act §26C.2) | MUST | WITHIN | 30 days |
| (Act §26C.2) | UPON | becoming aware a data breach may have occurred | |
| (Act §26C.1) | IF | the data breach occurs after the date of commencement of PDP(A) | A 2020 §13 |
| (Act §26C.2) | → | assess | if it is a Notifiable Data Breach |
| | HENCE | <i>Notification</i> | |

$\forall o . \text{organisation}(o) \wedge \neg \text{publicAgency}(o)$

Natural L4 to logic

| | STRUCTURED | UNSTRUCTURED |
|--------------|------------|---|
| (Act §26C.2) | EVERY | Organisation |
| (Act §4.1.c) | WHO | is not a Public Agency |
| (Act §26C.2) | MUST | WITHIN 30 days |
| (Act §26C.2) | UPON | becoming aware a data breach may have occurred |
| (Act §26C.1) | IF | the data breach occurs after the date of commencement of PDP(A)A 2020 §13 |
| (Act §26C.2) | → | assess if it is a Notifiable Data Breach |
| | HENCE | Notification |

$\forall o . \text{organisation}(o) \wedge \neg \text{publicAgency}(o)$

Natural L4 to logic

| STRUCTURED | UNSTRUCTURED |
|---|---|
| EVERY WHO | Organisation is not a Public Agency |
| <ul style="list-style-type: none">• Shape of formula• Quantification• Arguments to predicates | <ul style="list-style-type: none">• Names of predicates• Arities of predicates• Negation, ... |

$\forall o . \text{organisation}(o) \wedge \neg \text{publicAgency}(o)$

Natural L4 to logic

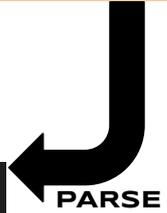
| STRUCTURED | UNSTRUCTURED |
|---|---|
| EVERY | Organisation |
| WHO | is not a Public Agency |
| UPON | becoming aware a data breach may have occurred |
| <ul style="list-style-type: none">• Shape of formula• Quantification• Arguments to predicates | <ul style="list-style-type: none">• Names of predicates• Arities of predicates• Negation, ...• Arguments to predicates |

$$\forall o, d. \text{organisation}(o) \wedge \neg \text{publicAgency}(o) \\ \wedge \text{dataBreach}(d) \wedge \text{becomeAwareOccur}(o, d)$$

Parsed into a Haskell datatype

| § Assessment | |
|--------------|--|
| (Act §26C.2) | EVERY Organisation ("you") |
| (Act §4.1.c) | WHO is not a Public Agency |
| (Act §26C.2) | MUST WITHIN 30 days |
| (Act §26C.2) | UPON becoming aware a data breach may have occurred |
| (Act §26C.1) | IF the data breach occurs after the date of commencement |
| (Act §26C.2) | → assess if it is a Notifiable Data Breach |
| | HENCE Notification |

NATURAL L4

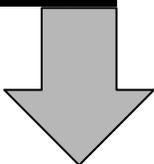


```
Regulative
{ subj = Leaf
  (
    ( "organisation" :| []
      , Nothing
    ) :| []
  )
, keyword = Every
, who = Just
  ( Leaf
    (
      ( "is not a public agency" :| []
        , Nothing
      ) :| []
    )
  )
, cond = Just
  ( Leaf
    (
      ( "the data breach occurs after the date of commencement"
        , Nothing
      ) :| []
    )
  )
, deontic = DMust
, action = Leaf
  (
    ( "assess" :| [ "if it is a notifiable data breach" ]
      , Nothing
    ) :| []
  )
, temporal = Just
  ( TemporalConstraint TBefore 30 "days" )
, hence = Just
  ( RuleAlias "Notification" )
, lest = Nothing
, rlabel = Nothing
, lsource = Nothing
, srcref = Nothing
, upon =
  [ Leaf
    (
      ( "becoming aware that a data breach may have occurred"
        , Nothing
      ) :| []
    )
  ]
, orig = []
}
```

STRINGS

Haskell datatype parsed with UD+GF

```
, who = Just
  ( Leaf
    ( "is not a public agency" UD PARSE
      , Nothing
    ) :| []
  )
)
```



ud, UD tree in CoNLLU format:

| | | | | | | |
|---|--------|--------|------|---|---|--------|
| 1 | is | be | AUX | Mood=Ind Number=Sing Person=3 Tense=Pres VerbForm=Fin | 5 | cop |
| 2 | not | not | PART | _ | 5 | advmod |
| 3 | a | a | DET | Definite=Ind PronType=Art | 5 | det |
| 4 | public | public | ADJ | Degree=Pos | 5 | amod |
| 5 | agency | agency | NOUN | Number=Sing | 0 | root |

Haskell datatype parsed with UD+GF

```
, who = Just
  ( Leaf
    ( "is not a public agency"
      , Nothing
      ) :| []
    )
  )
```

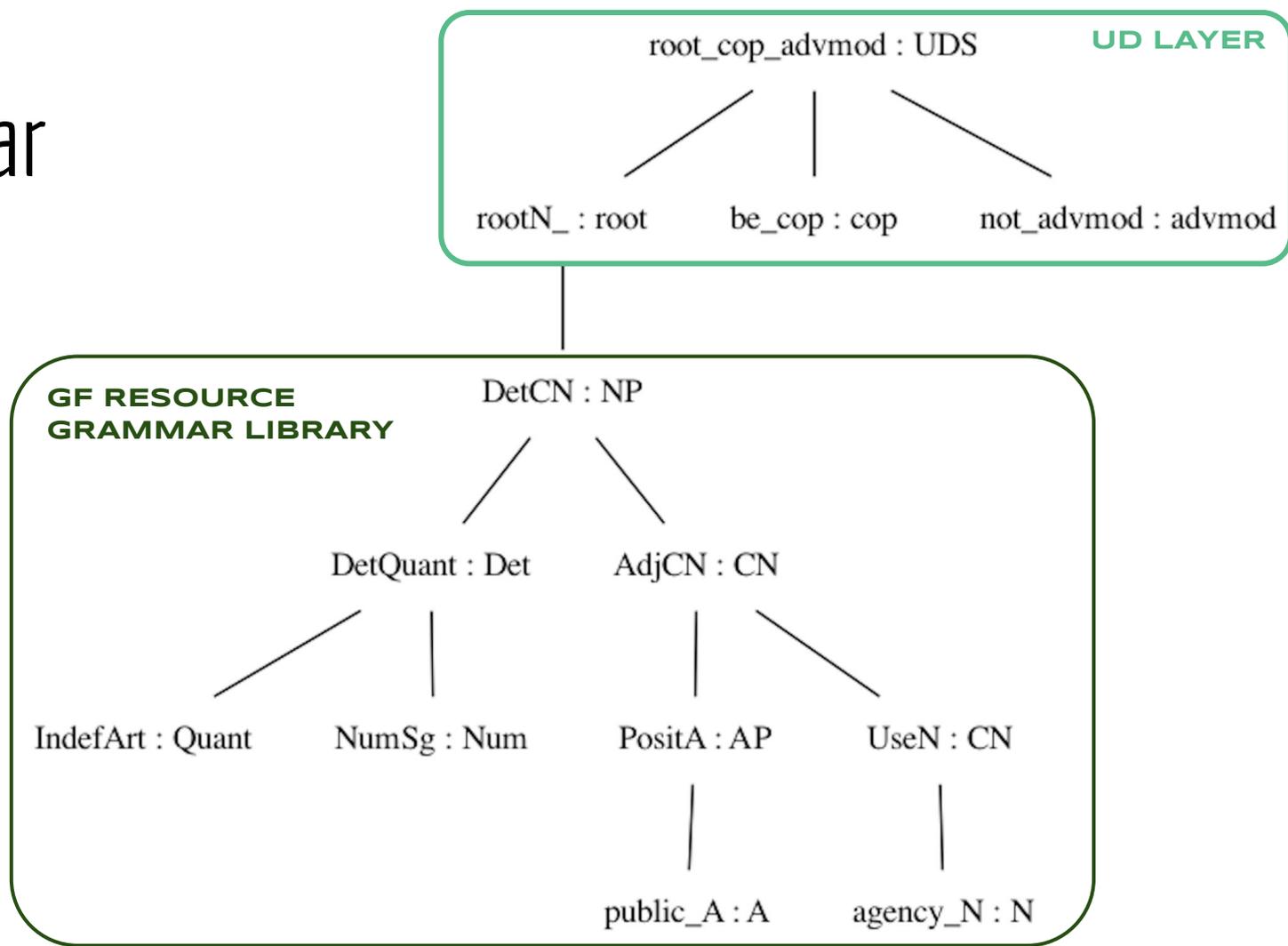
UD PARSE →

```
# ud, UD tree in CoNLLU format:
1  is      be      AUX ... 5  cop
2  not     not     PART ... 5  advmod
3  a       a       DET ... 5  det
4  public  public  ADJ ... 5  amod
5  agency  agency  NOUN ... 0  root
```

UD2GF

```
# GF tree
root_cop_advmod
  (rootN_ (DetCN (DetQuant IndefArt NumSg)
                (AdjCN (PositA public_A) (UseN agency_N))))
be_cop
not_advmod
```

GF grammar



GF grammar: categories

- Translate UD labels to GF categories

| | Nominals | Clauses | Modifier words | Function Words |
|--|---|---|--|--|
| Core arguments | nsubj obj iobj | csubj ccomp xcomp | | |
| Non-core dependents | obl vocative expl dislocated | advcl | advmod* discourse | aux cop mark |
| Nominal dependents | nmod appos nummod | acl | amod | det clf case |
| Coordination | MWE | Loose | Special | Other |
| conj cc | fixed flat compound | list parataxis | orphan goeswith reparandum | punct root dep |

```
abstract UDCat = {
  cat
```

```
  UDS ;
  acl ;
  aclRelcl ;
  advcl ;
  advmod ;
  advmodEmph ;
  advmodLmod ;
  amod ;
  appos ;
  aux ;
  auxPass ;
  case_ ;
  cc ;
  ccPreconj ;
  ccomp ;
  clf ;
  compound ;
  compoundLvc ;
  compoundPrt ;
  compoundRedup ;
  compoundSvc ;
  conj ;
```

```
  cop ;
  csubj ;
  csubjPass ;
  dep ;
  det ;
  detNumgov ;
  detNummod ;
  detPoss ;
  discourse ;
  dislocated ;
  expl ;
  explImpers ;
  explPass ;
  explPv ;
  fixed ;
  flat ;
  flatForeign ;
  flatName ;
  goeswith ;
  iobj ;
  list ;
  mark ;
```

```
  nmod ;
  nmodPoss ;
  nmodTmod ;
  nsubj ;
  nsubjPass ;
  nummod ;
  nummodGov ;
  obj ;
  oblAgent ;
  oblArg ;
  oblLmod ;
  oblTmod ;
  orphan ;
  parataxis ;
  punct ;
  reparandum ;
  root ;
  vocative ;
  xcomp ;
```

GF grammar: categories

- Translate UD labels to GF categories

| | Nominals | Clauses | Modifier words | Function Words |
|---------------------|---|------------------------|-------------------|----------------------|
| Core arguments | nsubj obj iobj | root csubj ccomp | | |
| Non-core dependents | obl nmod:poss nmod advcl:located | advcl | advmod* advmod | aux cop aux:is |
| Nominal dependents | nmod nmod:poss nummod | acl | advcl | det clf case |
| Coordination | conj | loose | special | other |
| cc | fixed flat foreign | flat foreign | advcl advmod | aux cop dep |

```
abstract UDCat = {
  cat
  UDS ;
  acl ;
  aclRelcl ;
  advcl ;
  advmod ;
  advmodEmph ;
  advmodLmod ;
  amod ;
  appos ;
  aux ;
  auxPass ;
  case_ ;
  cc ;
  ccPreconj ;
  ccomp ;
  clf ;
  compound ;
  compoundLvc ;
  compoundPrt ;
  compoundRedup ;
  compoundSvc ;
  conj ;
```

```
cop ;
csubj ;
csubjPass ;
dep ;
det ;
detNumgov ;
detNummod ;
detPoss ;
discourse ;
dislocated ;
expl ;
explImpers ;
explPass ;
explPv ;
fixed ;
flat ;
flatForeign ;
flatName ;
goeswith ;
iobj ;
list ;
mark ;
```

```
nmod ;
nmodPoss ;
nmodTmod ;
nsubj ;
nsubjPass ;
nummod ;
nummodGov ;
obj ;
obl ;
oblAgent ;
oblArg ;
oblLmod ;
oblTmod ;
orphan ;
parataxis ;
punct ;
reparandum ;
root ;
vocative ;
xcomp ;
```

- Map existing GF categories to the new UD-label categories

```
fun
-- UD roots can be many GF cats
rootV_ : VP -> root ;
rootA_ : AP -> root ;
rootN_ : NP -> root ;
rootAdv_ : Adv -> root ; -- within 30 days

-- GF NPs can have many UD labels
nsubj_ : NP -> nsubj ; -- lexical cat can be NOUN, DET, PRON, ...
obj_ : NP -> obj ; -- but all become eventually NPs in GF
iobj_ : NP -> iobj ;

-- Some UD words are syncategorematic in GF
be_cop : cop ;
it_expl : expl ; -- render [it] unlikely that ...
not_advmod : advmod ;
```

GF grammar: functions

- Extract sentence patterns from corpus

Act §26.2

the cat sleeps

observance is mandatory

she is here

*render unlikely that the breach will result in
significant harm*

is not a public agency

GF grammar: functions

- Extract sentence patterns from corpus

Act §26.2

root_nummod

the cat sleeps

root_nsubj

observance is mandatory

root_nsubj_cop

she is here

render unlikely that the breach will result in significant harm

root_xcomp_ccomp

is not a public agency

root_cop_advmod

GF grammar: functions

- Extract sentence patterns from corpus

Act §26.2

root_nummod

the cat sleeps

root_nsubj

observance is mandatory

root_nsubj_cop

she is here

render unlikely that the breach will result in significant harm

root_xcomp_ccomp

is not a public agency

root_cop_advmod

GF grammar: functions

- Extract sentence patterns from corpus

Act §26.2

root_nummod

the cat sleeps

root_nsubj

observance is mandatory

root_nsubj_cop

she is here

render unlikely that the breach will result in significant harm

root_xcomp_ccomp

is not a public agency

root_cop_advmod

```
fun root_acl : root -> acl -> UDS ;
fun root_acl_nmod : root -> acl -> nmod -> UDS ;
fun root_advcl : root -> advcl -> UDS ;
fun root_advmod : root -> advmod -> UDS ;
fun root_advmod_advmodobl : root -> advmod -> advmod -> obl -> UDS ;
fun root_advmod_amod : root -> advmod -> amod -> UDS ;
fun root_advmod_nsubj_copobl : root -> advmod -> nsubj -> cop -> obl -> UDS ;
fun root_advmod_xcomp : root -> advmod -> xcomp -> UDS ;
fun root_amod : root -> amod -> UDS ;
fun root_amod_nmod : root -> amod -> nmod -> UDS ;
fun root_appos : root -> appos -> UDS ;
fun root_appos_advmod : root -> appos -> advmod -> UDS ;
fun root_auxPass : root -> auxPass -> UDS ;
fun root_case : root -> case -> UDS ;
fun root_case_amod : root -> case -> amod -> UDS ;
fun root_case_amod_amod : root -> case -> amod -> amod -> UDS ;
fun root_case_amod_conj : root -> case -> amod -> conj -> UDS ;
fun root_case_compound : root -> case -> compound -> UDS ;
fun root_case_det : root -> case -> det -> UDS ;
fun root_case_det_amod : root -> case -> det -> amod -> UDS ;
fun root_case_det_compound_conj : root -> case -> det -> compound -> conj -> UDS ;
fun root_case_det_nmod : root -> case -> det -> nmod -> UDS ;
fun root_case_nummod : root -> case -> nummod -> UDS ;
fun root_case_nummod_acl : root -> case -> nummod -> acl -> UDS ;
fun root_case_nummod_nmod : root -> case -> nummod -> nmod -> UDS ;
fun root_cc : root -> cc -> UDS ;
fun root_cc_aux_cop_det_nmod : root -> cc -> aux -> cop -> det -> nmod -> UDS ;
fun root_cc_conj : root -> cc -> conj -> UDS ;
fun root_cc_cop_xcomp : root -> cc -> cop -> xcomp -> UDS ;
fun root_cc_det_nmod : root -> cc -> det -> nmod -> UDS ;
fun root_cc_nmod : root -> cc -> nmod -> UDS ;
fun root_cc_obj : root -> cc -> obj -> UDS ;
fun root_ccomp : root -> ccomp -> UDS ;
fun root_compound : root -> compound -> UDS ;
fun root_compoundPrt_compoundPrt : root -> compoundPrt -> compoundPrt -> UDS ;
fun root_compound_acl : root -> compound -> acl -> UDS ;
fun root_compound_amod : root -> compound -> amod -> UDS ;
fun root_compound_appos : root -> compound -> appos -> UDS ;
fun root_compound_compound : root -> compound -> compound -> UDS ;
fun root_compound_compound_appos : root -> compound -> compound -> appos -> UDS ;
fun root_compound_compound_conj : root -> compound -> compound -> conj -> UDS ;
fun root_compound_conj_acl : root -> compound -> conj -> acl -> UDS ;
fun root_compound_flat : root -> compound -> flat -> UDS ;
fun root_conj : root -> conj -> UDS ;
fun root_conj_acl : root -> conj -> acl -> UDS ;
fun root_conj_appos : root -> conj -> appos -> UDS ;
fun root_conj_case : root -> conj -> case -> UDS ;
fun root_conj_nmod : root -> conj -> nmod -> UDS ;
fun root_conj_parataxis : root -> conj -> parataxis -> UDS ;
fun root_cop : root -> cop -> UDS ;
fun root_cop_advmod : root -> cop -> advmod -> UDS ;
fun root_cop_conj : root -> cop -> conj -> UDS ;
fun root_cop_det_compound_amod : root -> cop -> det -> compound -> amod -> UDS ;
fun root_cop_det_nmod : root -> cop -> det -> nmod -> UDS ;
fun root_csobj : root -> csobj -> UDS ;
fun root_csobj_aux_aux : root -> csobj -> aux -> aux -> UDS ;
fun root_det : root -> det -> UDS ;
fun root_det_acl : root -> det -> acl -> UDS ;
fun root_det_aclRelcl : root -> det -> aclRelcl -> UDS ;
fun root_det_aclRelcl_nmod : root -> det -> aclRelcl -> nmod -> UDS ;
fun root_det_advmod : root -> det -> advmod -> UDS ;
fun root_det_amod : root -> det -> amod -> UDS ;
fun root_det_amod_aclRelcl : root -> det -> amod -> aclRelcl -> UDS ;
fun root_det_amod_aclRelcl_nmod : root -> det -> amod -> aclRelcl -> nmod -> UDS ;
fun root_det_amod_amod_acl_nmod : root -> det -> amod -> amod -> acl -> nmod -> UDS ;
fun root_det_amod_nmod : root -> det -> amod -> nmod -> UDS ;
fun root_nummod_nsubjPass_nsubjPass_auxPass_cc : root -> nummod -> nsubjPass -> nsubjPass -> auxPass -> cc -> UDS ;
fun root_nummod_obl : root -> nummod -> obl -> UDS ;
fun root_nummod_obl_cc : root -> nummod -> obl -> cc -> UDS ;
fun root_obj : root -> obj -> UDS ;
fun root_obj_ccomp : root -> obj -> ccomp -> UDS ;
fun root_obj_nmod : root -> obj -> nmod -> UDS ;
fun root_obl : root -> obl -> UDS ;
fun root_obl_appos : root -> obl -> appos -> UDS ;
fun root_obl_aux : root -> obl -> aux -> UDS ;
fun root_obl_case : root -> obl -> case -> UDS ;
fun root_obl_obj : root -> obl -> obj -> UDS ;
fun root_obl_obj : root -> obl -> obj -> UDS ;
fun root_obl_obj_acl_cc : root -> obl -> obj -> acl -> UDS ;
fun root_obl_xcomp : root -> obl -> xcomp -> UDS ;
fun root_only : root -> UDS ;
fun root_parataxis : root -> parataxis -> UDS ;
fun root_xcomp : root -> xcomp -> UDS ;
fun root_xcomp_ccomp : root -> xcomp -> ccomp -> UDS ;
```

GF grammar: functions

- UD-functions take RGL trees as argument

Act §26.2

root_nummod

the cat sleeps

root_nsubj

observance is mandatory

root_nsubj_cop

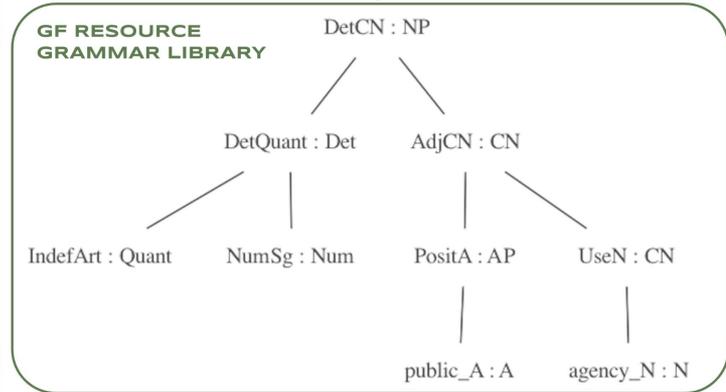
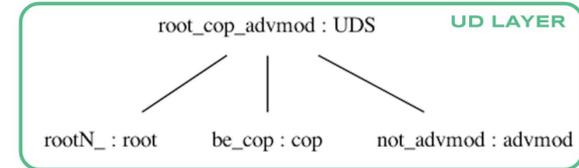
she is here

render unlikely that the breach will result in significant harm

root_xcomp_ccomp

is not a public agency

root_cop_advmod



GF grammar: functions

Act §26.2

root_nummod

the cat sleeps

root_nsubj

observance is mandatory

root_nsubj_cop

she is here

render unlikely that the breach will result in significant harm

root_xcomp_ccomp

is not a public agency

root_cop_advmod

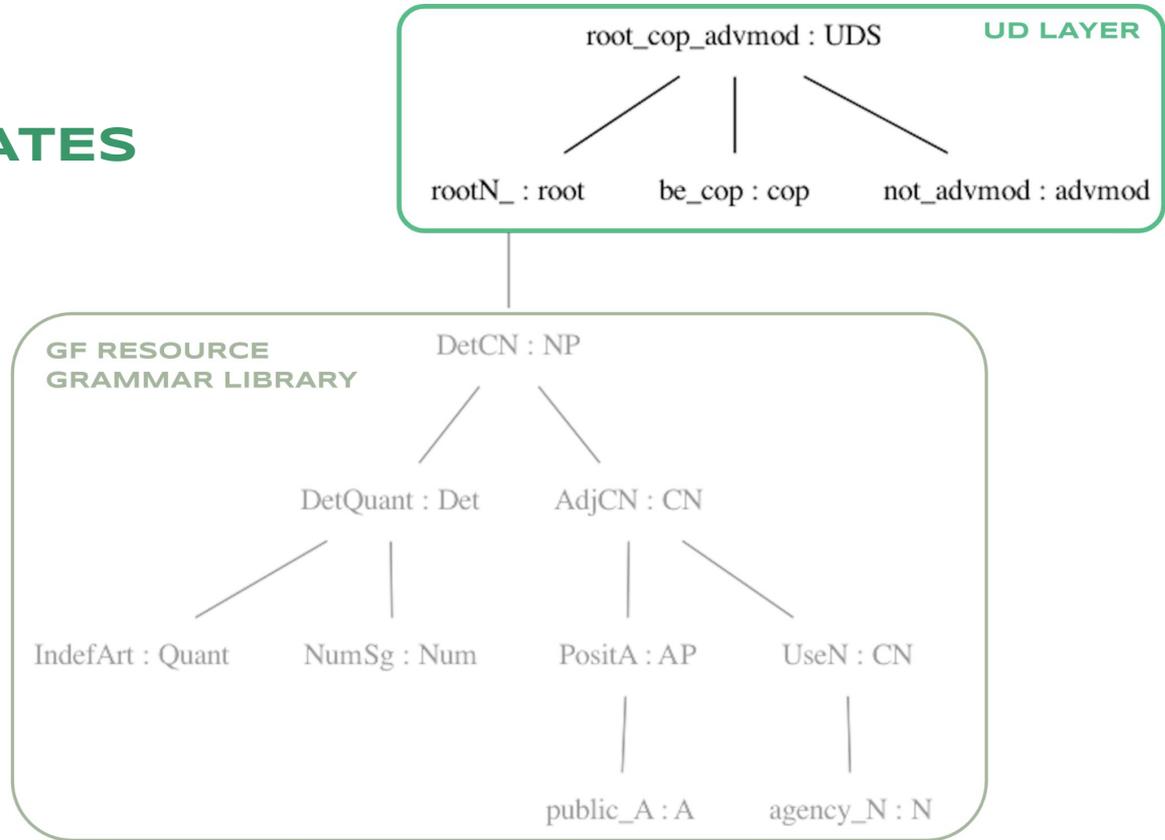
```
fun root_acl : root -> acl -> UDS ;
fun root_acl_nmod : root -> acl -> nmod -> UDS ;
fun root_advcl : root -> advcl -> UDS ;
fun root_advmod : root -> advmod -> UDS ;
fun root_advmod_advmodobl : root -> advmod -> advmod -> obl -> UDS ;
fun root_advmod_amod : root -> advmod -> amod -> UDS ;
fun root_advmod_nsubj_copobl : root -> advmod -> nsubj -> cop -> obl -> UDS ;
fun root_advmod_xcomp : root -> advmod -> xcomp -> UDS ;
fun root_amod : root -> amod -> UDS ;
fun root_amod_nmod : root -> amod -> nmod -> UDS ;
fun root_appos : root -> appos -> UDS ;
fun root_appos_advmod : root -> appos -> advmod -> UDS ;
fun root_auxPass : root -> auxPass -> UDS ;
fun root_case : root -> case -> UDS ;
fun root_case_amod : root -> case -> amod -> UDS ;
fun root_case_amod_amod : root -> case -> amod -> amod -> UDS ;
fun root_case_amod_conj_conj : root -> case -> amod -> conj -> conj -> UDS ;
fun root_case_compound : root -> case -> compound -> UDS ;
fun root_case_det : root -> case -> det -> UDS ;
fun root_case_det_amod : root -> case -> det -> amod -> UDS ;
fun root_case_det_compound_conj : root -> case -> det -> compound -> conj -> UDS ;
fun root_case_det_nmod : root -> case -> det -> nmod -> UDS ;
fun root_case_nummod : root -> case -> nummod -> UDS ;
fun root_case_nummod_acl : root -> case -> nummod -> acl -> UDS ;
fun root_case_nummod_nmod : root -> case -> nummod -> nummod -> UDS ;
fun root_cc : root -> cc -> UDS ;
fun root_cc_aux_cop_det_nmod : root -> cc -> aux -> cop -> det -> nmod -> UDS ;
fun root_cc_conj : root -> cc -> conj -> UDS ;
fun root_cc_cop_xcomp : root -> cc -> cop -> xcomp -> UDS ;
fun root_cc_det_nmod : root -> cc -> det -> nmod -> UDS ;
fun root_cc_nmod : root -> cc -> nmod -> UDS ;
fun root_cc_obj : root -> cc -> obj -> UDS ;
fun root_ccomp : root -> ccomp -> UDS ;
fun root_cop : root -> cop -> UDS ;
fun root_cop_advmod : root -> cop -> advmod -> UDS ;
fun root_cop_conj_conj : root -> cop -> conj -> conj -> UDS ;
fun root_cop_det_compound_amod : root -> cop -> det -> compound -> amod -> UDS ;
fun root_cop_det_nmod : root -> cop -> det -> nmod -> UDS ;
fun root_csobj : root -> csobj -> UDS ;
fun root_csobj_aux_aux : root -> csobj -> aux -> aux -> UDS ;
fun root_det : root -> det -> UDS ;
fun root_det_acl : root -> det -> acl -> UDS ;
fun root_det_aclRelcl : root -> det -> aclRelcl -> UDS ;
fun root_det_aclRelcl_nmod : root -> det -> aclRelcl -> nmod -> UDS ;
fun root_det_advmod : root -> det -> advmod -> UDS ;
fun root_det_amod : root -> det -> amod -> UDS ;
fun root_det_amod_aclRelcl : root -> det -> amod -> aclRelcl -> UDS ;
fun root_det_amod_aclRelcl_nmod : root -> det -> amod -> aclRelcl -> nmod -> UDS ;
fun root_det_amod_nmod : root -> det -> amod -> nmod -> UDS ;
fun root_det_nmod : root -> det -> nmod -> UDS ;
fun root_nummod_nsubjPass_nsubjPass_auxPass_cc : root -> nummod -> nsubjPass -> nsubjPass -> auxPass -> cc -> UDS ;
fun root_nummod_obl : root -> nummod -> obl -> UDS ;
fun root_nummod_obl_cc : root -> nummod -> obl -> cc -> UDS ;
fun root_obj : root -> obj -> UDS ;
fun root_obj_ccomp : root -> obj -> ccomp -> UDS ;
fun root_obj_nmod : root -> obj -> nmod -> UDS ;
fun root_obl : root -> obl -> UDS ;
fun root_obl_appos : root -> obl -> appos -> UDS ;
fun root_obl_aux : root -> obl -> aux -> UDS ;
fun root_obl_case : root -> obl -> case -> UDS ;
fun root_obl_obj : root -> obl -> obj -> UDS ;
fun root_obl_obl : root -> obl -> obl -> UDS ;
fun root_obl_obl_acl_cc : root -> obl -> obl -> acl -> UDS ;
fun root_obl_xcomp : root -> obl -> xcomp -> UDS ;
fun root_only : root -> UDS ;
fun root_parataxis : root -> parataxis -> UDS ;
fun root_xcomp : root -> xcomp -> UDS ;
fun root_xcomp_ccomp : root -> xcomp -> ccomp -> UDS ;
```

No linearisation needed

Storage for RGL trees

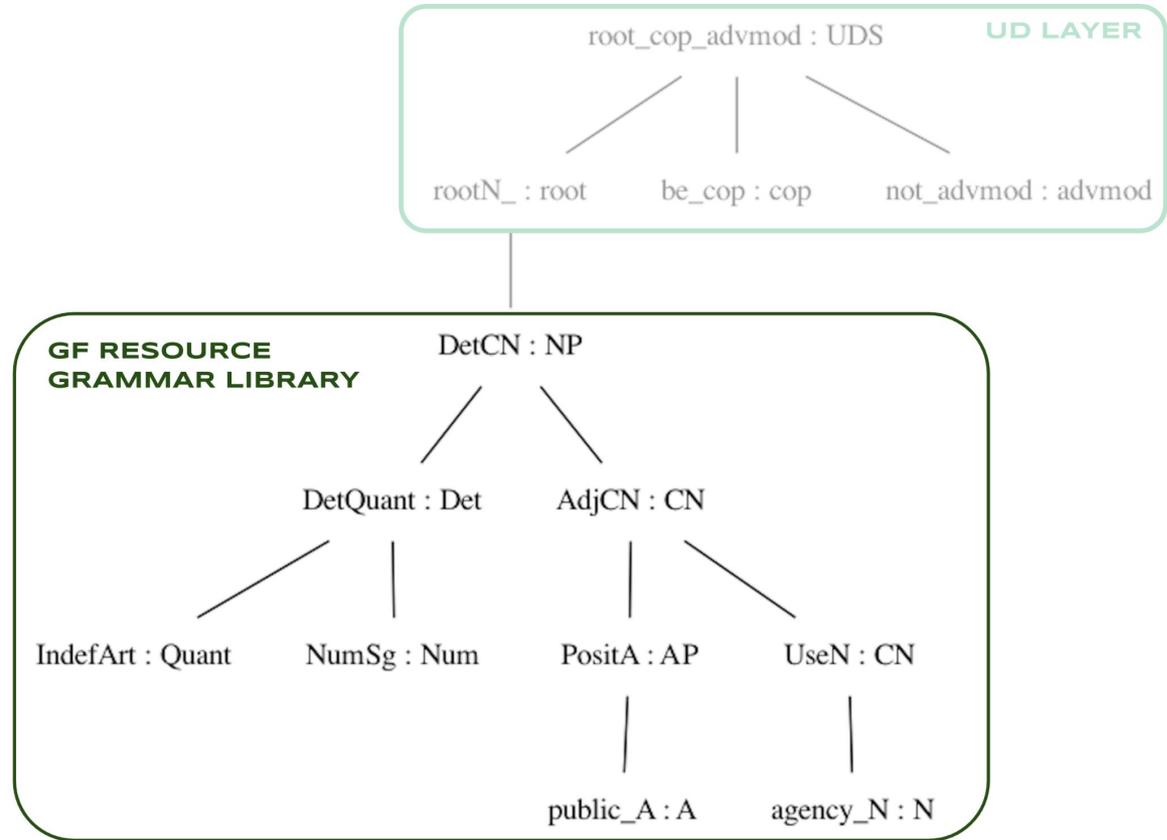
GF grammar

EXTRACT PREDICATES



GF grammar

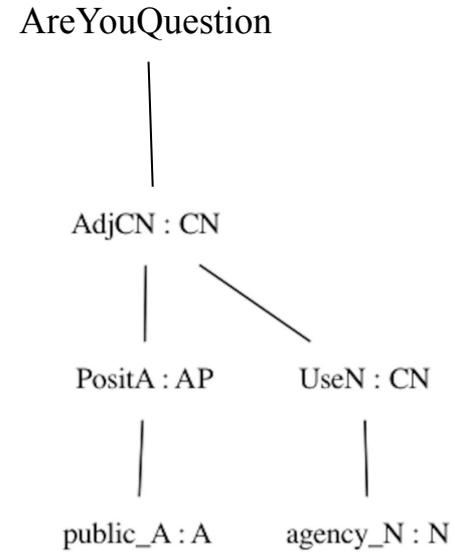
NLG



GF grammar

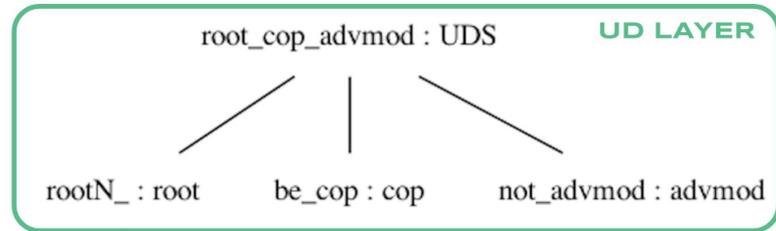
NLG

"are you a public agency?"



Predicate extraction

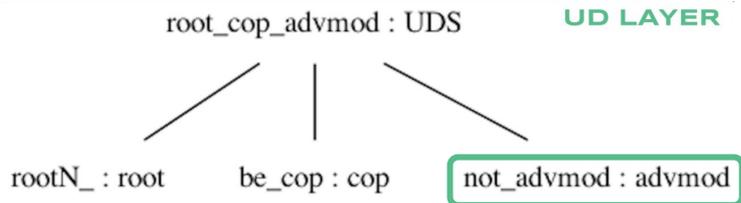
- Pattern match UD-trees



```
mkPredicate :: Gf (Tree a) => Tree a -> Predicate
mkPredicate (GrootN_ x) = Unary $ headNP x
mkPredicate (GrootV_ x) = Unary $ headVP x
mkPredicate (Groot_cop_advmod root _ Gnot_advmod) = Not $ mkPredicate root
```

Predicate extraction

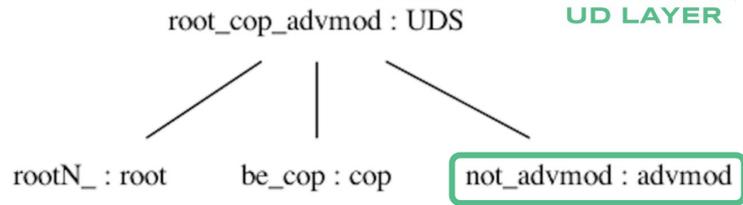
- Pattern match UD-trees



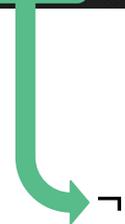
```
mkPredicate :: Gf (Tree a) => Tree a -> Predicate
mkPredicate (GrootN_ x) = Unary $ headNP x
mkPredicate (GrootV_ x) = Unary $ headVP x
mkPredicate (Groot_cop_advmod root _ Gnot_advmod) = Not $ mkPredicate root
```

Predicate extraction

- Extract negation from UD-tree

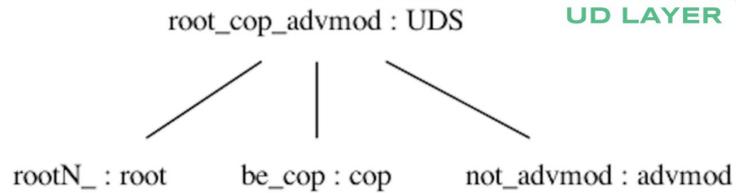


```
mkPredicate :: Gf (Tree a) => Tree a -> Predicate
mkPredicate (GrootN_ x) = Unary $ headNP x
mkPredicate (GrootV_ x) = Unary $ headVP x
mkPredicate (Groot_cop_advmod root _ Gnot_advmod) = Not $ mkPredicate root
```



Predicate extraction

- Pattern match root

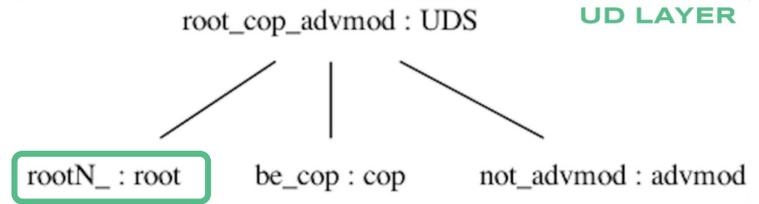


```
mkPredicate :: Gf (Tree a) => Tree a -> Predicate
mkPredicate (GrootN_ x) = Unary $ headNP x
mkPredicate (GrootV_ x) = Unary $ headVP x
mkPredicate (Groot_cop_advmod root _ Gnot_advmod) = Not $ mkPredicate root
```



Predicate extraction

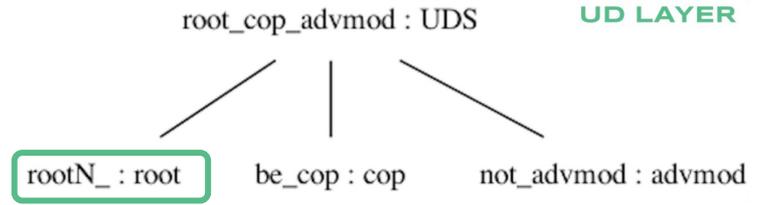
- Pattern match root



```
mkPredicate :: Gf (Tree a) => Tree a -> Predicate
mkPredicate (GrootN_ x) = Unary $ headNP x
mkPredicate (GrootV_ x) = Unary $ headVP x
mkPredicate (Groot_cop_advmod root _ Gnot_advmod) = Not $ mkPredicate root
```

Predicate extraction

- Extract predicate name



```
mkPredicate :: Gf (Tree a) => Tree a -> Predicate
mkPredicate (GrootN_ x) = Unary $ headNP x
mkPredicate (GrootV_ x) = Unary $ headVP x
mkPredicate (Groot_cop_advmod root (Groot_advmod)) = Not $ mkPredicate root

headCN :: GCN -> String
headCN (GUseN (Lexical n)) = n -- organization
headCN (GUseN (GCompoundN (Lexical n1) (Lexical n2))) =
  n1 `combineName` n2 -- dataBreach
headCN (GAdjCN ap cn) = -- publicAgency
  headAP ap `combineName` headCN cn
```

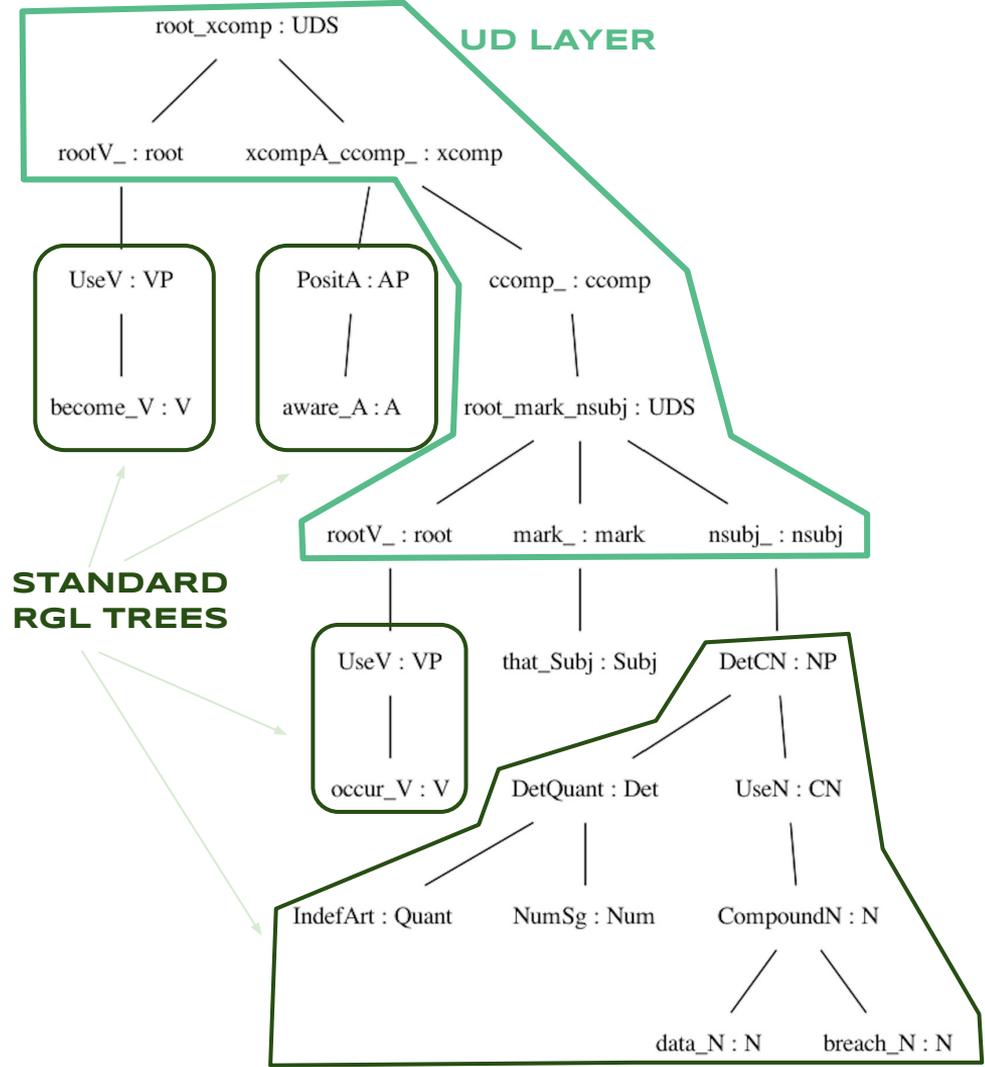
→ publicAgency

Negation?

- Logical negation, or part of the predicate?
 - $\neg\text{publicAgency}(x)$ vs. $\text{notPublicAgency}(x)$
 - For nested predicates, only latter feasible: “A knows that B is not a public agency”
- Good for NLG if predicates are simple
 - “are you a public agency” vs. “are you not a public agency”
 - But NLG and logic can be different!

Nested clauses

“becoming aware that a data breach occurs”



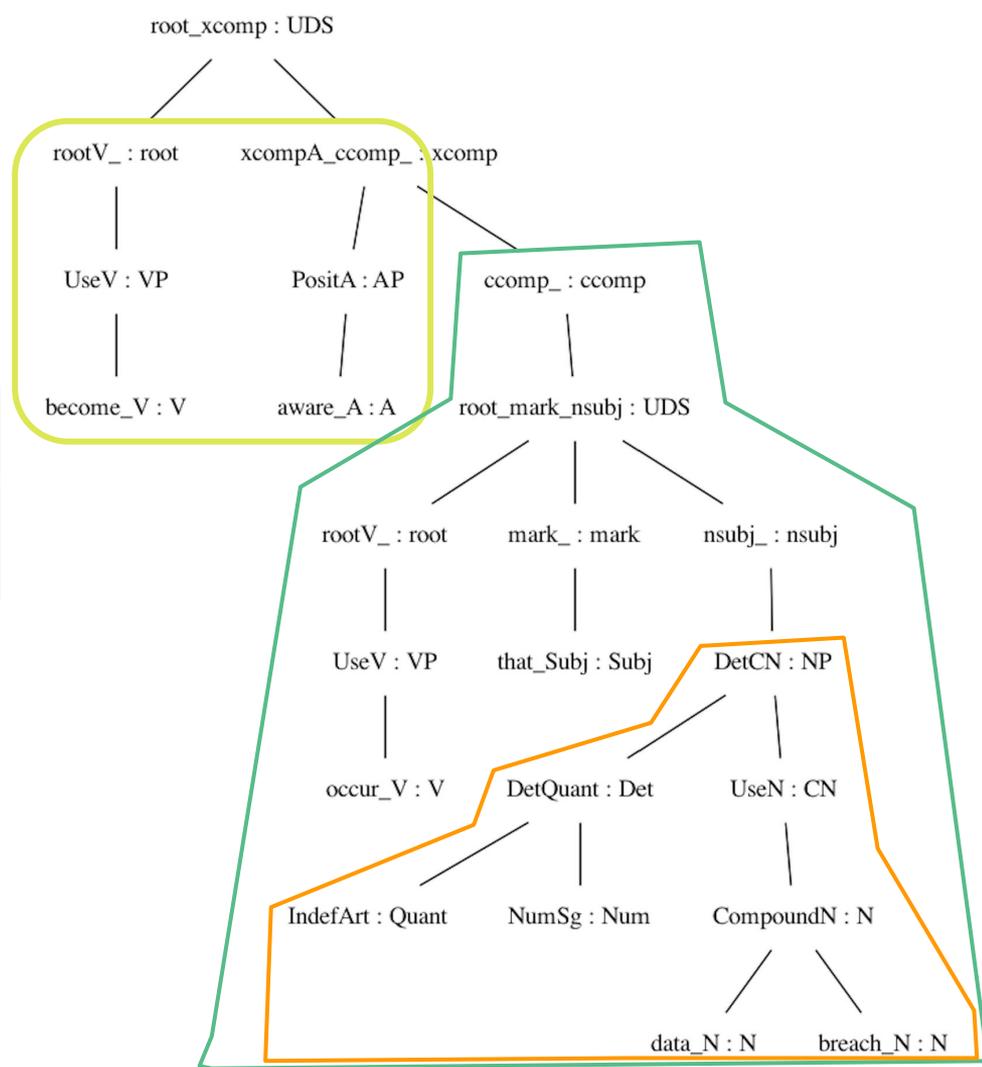
Nested clauses

```
, upon =  
  [ Leaf  
    (  
      ("becoming aware a data breach may have occurred"  
       Nothing  
       :| []  
      )  
    )  
  ] UPON ≈ EVENT PREREQ TRIGGER
```

If upon contains a VP, then its subject of upon field is the subj.

Whatever ccomp the upon might have, is a predicate about the world.

If upon contains a sentence with subject, then upon is also an arbitrary predicate about the world.



Nested clauses

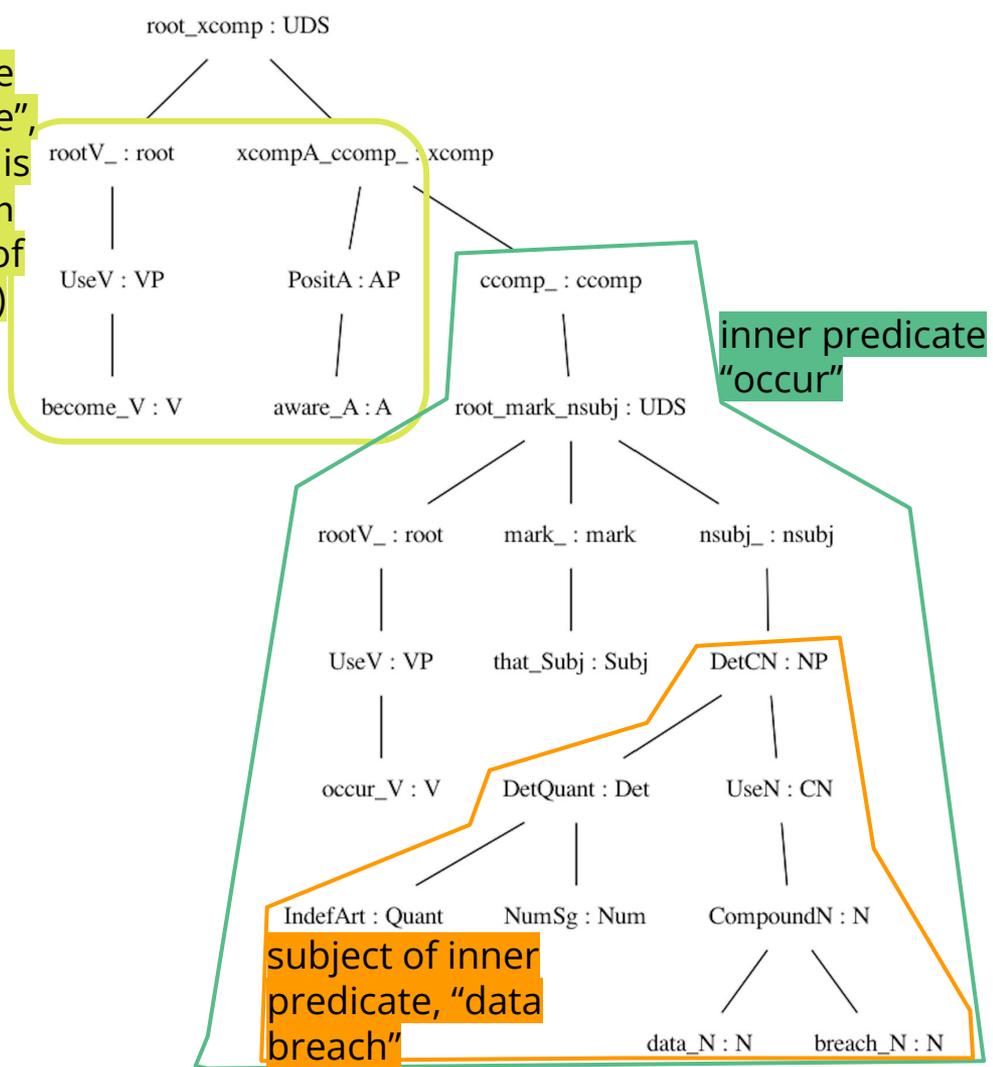
outer predicate
"become aware",
whose subject is
Organisation
(main subject of
the whole rule)

```
, upon =  
[ Leaf  
(  
  ("becoming aware a data breach may have occurred":  
   Nothing  
  )  
)  
]  
UPON ≈ EVENT PREREQ TRIGGER
```

If upon contains a VP, then its subject of upon field is the subj.

Whatever ccomp the upon might have, is a predicate about the world.

If upon contains a sentence with subject, then upon is also an arbitrary predicate about the world.



Nested clauses

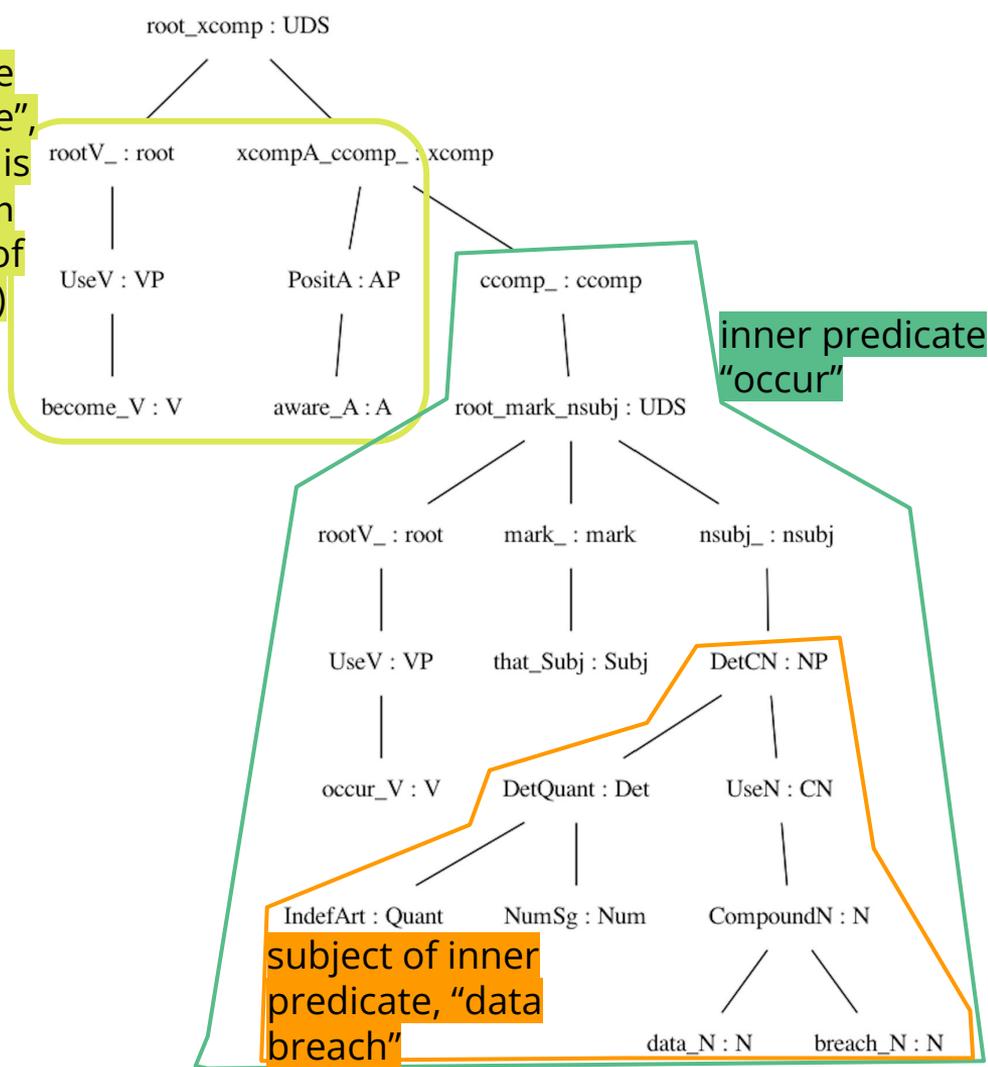
outer predicate
"become aware",
whose subject is
Organisation
(main subject of
the whole rule)

```

, upon =
[ Leaf
(
( "becoming aware a data breach may have occurred" :
Nothing
): | []
)
] UPON ≈ EVENT PREREQ TRIGGER
    
```

$\forall o, d. \text{dataBreach}(d) \wedge$

$\text{becomeAware}(\text{Occur}(o, d))$



Thank you

Code & more info at Singapore Management University's Centre for Computational Law:

<https://github.com/smucclaw>

Email me: inari.listenmaa@gmail.com