

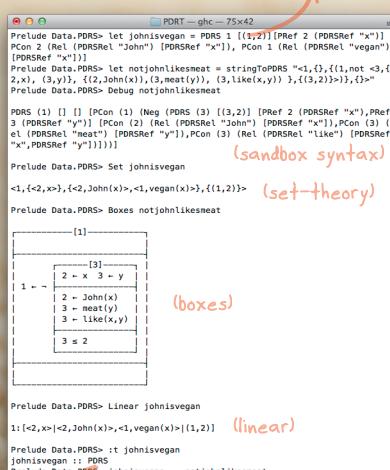
Projective Discourse Representation Theory (Venhuizen et al., 2013; in prep)

An extension of DRT (Kamp, 1981; Kamp & Reyle, 1993) with **projection variables** to account for projection phenomena, e.g., **presuppositions** (Venhuizen et al., 2013) and Potts' (2005) **conventional implicatures** (Venhuizen et al., 2014).

Projection variables indicate the interpretation site of semantic content, so all content stays *in situ* during composition.
 > Projected content is indicated by free pointers, or pointers bound to the label of a non-local, accessible PDRS.

Challenge: projection variables affect the **construction procedure** and definition of **accessibility** non-trivially.

input/output formats



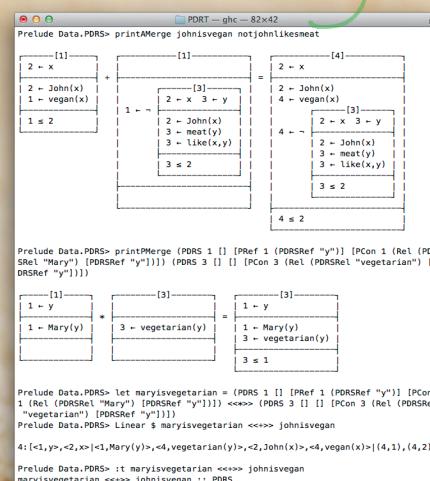
```

Prelude Data.PDRS> let johnsvegan = PDRS 1 [(1,2)] [Pref 2 (PDRSRef "x"), Pref 3 (PDRSRef "y"))] [PCon 2 (Rel (PDRSRef "John")) [PDRSRef "x"]], PCon 1 (Rel (PDRSRef "vegan")) [PDRSRef "x")]
Prelude Data.PDRS> let notjohnvegan = stringToPDRS "(1,1,{(1,net -> {(1,2), (3,y), {(2,1,John(x)),(3,meat(y)), (3,like(x,y))}, {(3,2)})}}))" 
Prelude Data.PDRS> Debug notjohnvegan
Prelude Data.PDRS> Set johnsvegan
<1,<2,x>,<2,John(x)>,<1,vegan(x)>,<(1,2)>
(sandbox syntax)
Prelude Data.PDRS> Boxes notjohnvegan
  
```

(boxes)

PDRT-SANDBOX: a Haskell library that implements PDRT and DRT

merging ↙

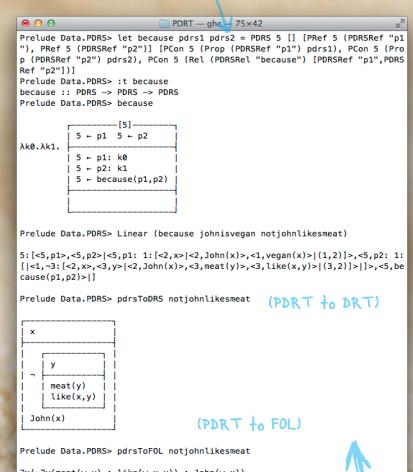


```

Prelude Data.PDRS> printAMerge johnsvegan notjohnvegan
  
```

A modular and flexible NLP library that incorporates machinery to build, combine, and translate structures from PDRT as well as DRT
 ... and much more!

lambda-abstractions ↙



```

Prelude Data.PDRS> let because pdr1 pdr2 = PDRS 5 [] [Pref 5 (PDRSRef "p1"), Pref 5 (PDRSRef "p2"))] [PCon 5 (Prop (PDRSRef "p1") pdr1), PCon 5 (Prop (PDRSRef "p2") pdr2)], PCon 1 (Rel (PDRSRef "because")) [PDRSRef "p1", PDRSRef "p2"]
Prelude Data.PDRS> it because because :: PDRS -> PDRS -> PDRS
Prelude Data.PDRS> because
  
```

Implementation

(variable binding and renaming!)

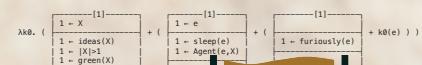
Accessibility of PDRT referents is defined using a **projection graph**.

Unresolved lambda-(P)DRSs (Muskens, 1996) as Haskell functions.

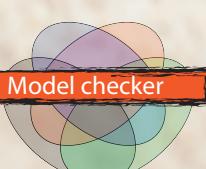
Various **input** (set-theoretic, sandbox syntax, Boxer's Prolog syntax; Bos, 2003), and **output** formats (*pretty printing* of (P)DRS representations, and non-recursive P-Tables).

Application

Forming an NLP toolchain together with a syntactic parser (à la C&C tools and Boxer; Curran et al., 2007), or reasoning about semantic representations using a theorem prover or model checker.



Model checker



translations ↗

