Andreas van Cranenburgh 0440949
Structures for Semantics, assignment 2,
University of Amsterdam, March 2010

Exercise 1

We have:

- every: $\langle\langle e,t\rangle,t\rangle$ $\lambda P\lambda Q\forall x[P(x) \to Q(x)]$
- woman: $\langle e,t\rangle$
- loves: $\langle e,\langle e,t\rangle\rangle$
- john, mary: e
- or: $\langle e,\langle e,\langle\langle e,t\rangle,t\rangle\rangle\rangle$ $\lambda x\lambda y\lambda P.(P(x) \vee P(y))$

We derive the following readings:

1. $(every(woman))(\lambda x.(AR2(loves)(x)(or(john)(mary))))$
   $\Rightarrow \forall x[woman(x) \to (loves(x, mary) \vee loves(x, john))]$
2. $or(john)(mary)((AR1(loves))(every(woman)))$
   $\Rightarrow \forall x[woman(x) \to loves(x, mary)] \vee \forall x[woman(x) \to loves(x, john)]$

Exercise 2

Imagine two worlds which disagree as to whether counting should start at zero or at one (they have not yet reached the compromise of starting at 0.5). This yields a counter-model, exploiting the fact that constants are non-rigid, so that they can refer to different values in the meta-language (right hand side):

$I(R)(w_0) = \{a\}$
$I(a)(w_0)(w_0) = 90$
$I(a)(w_0)(w_1) = 90$
$I(p)(w_0)(w_0) = 90$
$I(p)(w_0)(w_1) = 91$
$I(90)(w_0) = 90$

$I(R)(w_1) = \{a\}$
$I(a)(w_1)(w_0) = 91$
$I(a)(w_1)(w_1) = 91$
$I(p)(w_1)(w_0) = 90$
$I(p)(w_1)(w_1) = 91$
$I(90)(w_1) = 91$

Exercise 3

(a)

(4a) $\forall x[buy(mary, x) \to want(mary, \wedge(inexpensive(x) \wedge coat(x)))]$
(4b) $\exists x[coat(x) \wedge want(mary, \wedge(buy(mary, x) \wedge inexpensive(x))]$

(b)

Representation (5b) can be translated into Ty2 in the following manner:
$\lambda P(W_a(\lambda a.\exists x(P(x) \wedge B_a(x)(m_a)))(m_a))(I_a)$
Alpha-conversion:
$\lambda P(W_a(\lambda v.\exists x(P(x) \wedge B_v(x)(m_v)))(m_a))(I_a)$
Beta-conversion:
$W_a(\lambda v.\exists x(I_a(x) \wedge B_v(x)(m_v)))(m_a)$
This last expression is equivalent to (5a), QED.

Representation (5c) can be translated into Ty2 in the following manner:
$\lambda Q(W_a(\lambda a.(\exists x(Q(a)(x) \wedge B_a(x)(m_a))))(m_a))(\lambda a I_a)$
Alpha-conversion:
$\lambda Q(W_a(\lambda v.(\exists x(Q(v)(x) \wedge B_v(x)(m_v))))(m_a))(\lambda a I_a)$
Beta-conversion:
$W_a(\lambda v.(\exists x(I_v(x) \wedge B_v(x)(m_v))))(m_a)$
This last expression is not equivalent to (5a), because this expression has I_v instead of I_a, QED.
Representation (5d) can be translated into Ty2 in the following manner:
$\exists Q(Q(a) = I_a \wedge W_a(\lambda a.(\exists x(Q_a(x) \wedge B_a(x)(m))))(m_a))$
Alpha-conversion:
$\exists Q(Q(a) = I_a \wedge W_a(\lambda v.(\exists x(Q_v(x) \wedge B_v(x)(m_v))))(m_a))$
This last expression is clearly not equivalent to (5a), QED.

Exercise 4

We will derive the following readings:

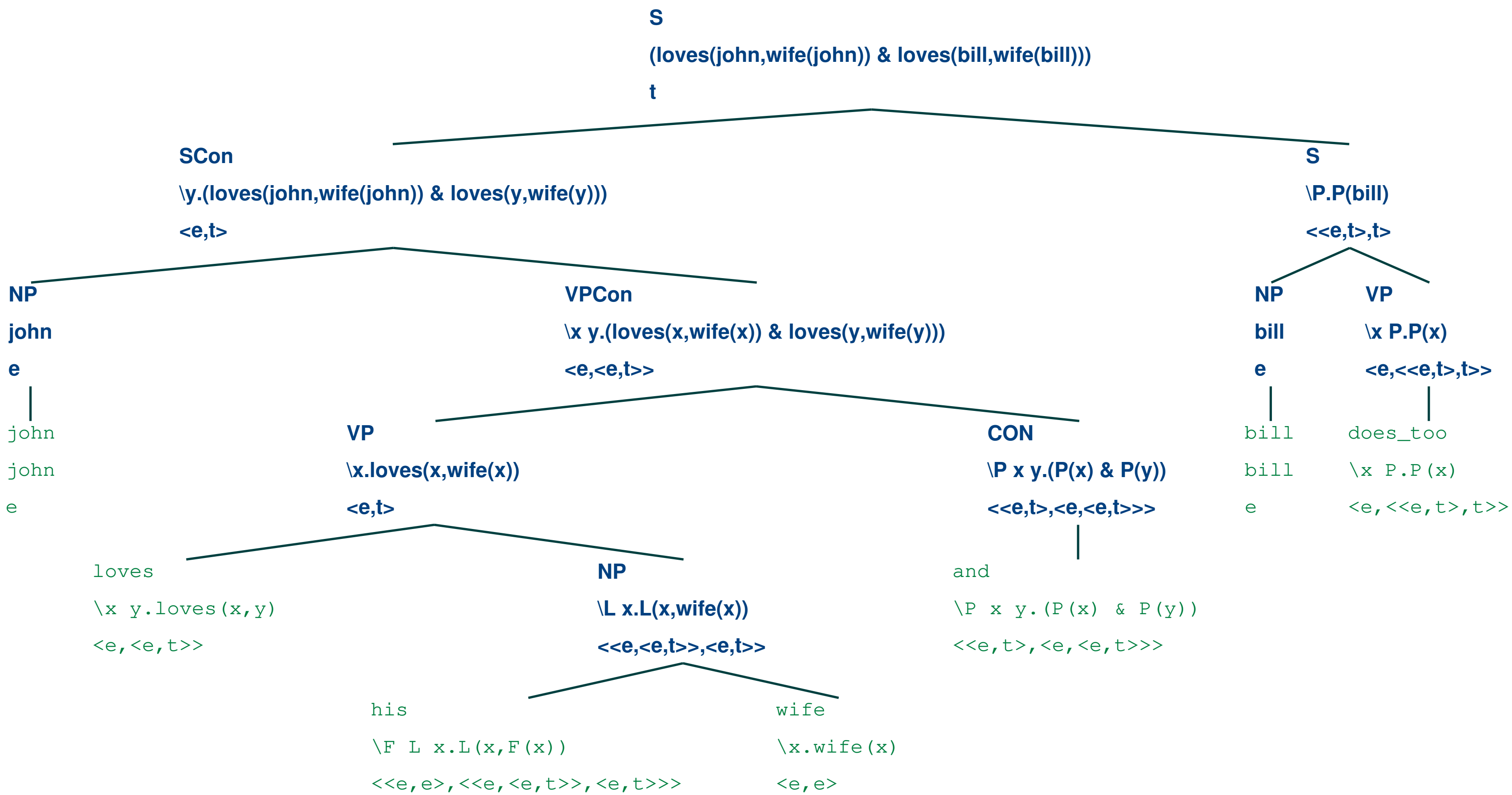
1. $loves(john, wife(john)) \wedge loves(bill, wife(bill))$
2. $loves(john, wife(john)) \wedge loves(bill, wife(john))$

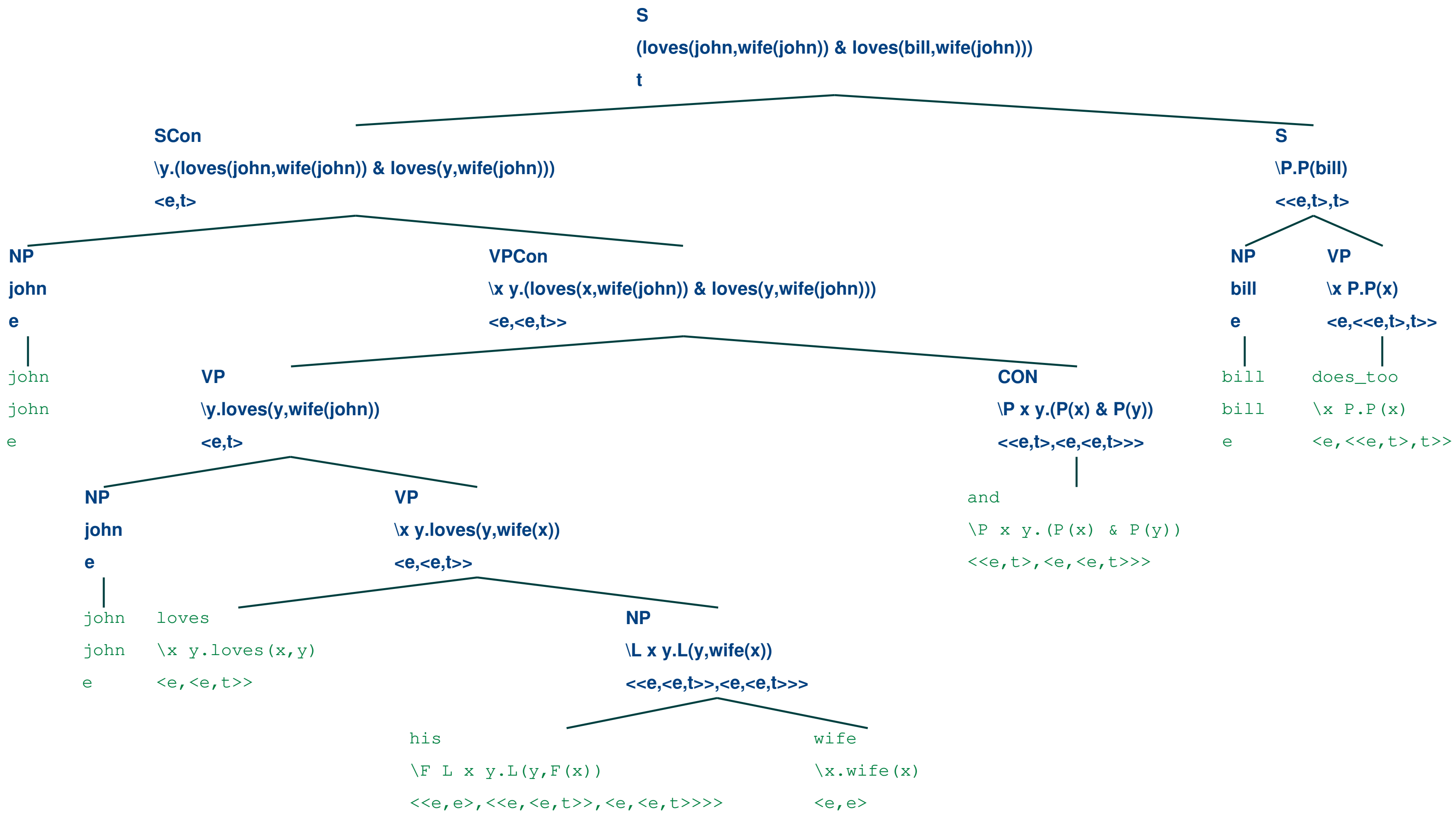
The lexicon:
john: $john$ e
loves: $\lambda xy.loves(x, y)$ $\langle e, \langle e, t\rangle\rangle$
his: $\lambda FLx.L(x, F(x))$ $\langle\langle e, e\rangle, \langle\langle e, \langle e, t\rangle\rangle, \langle e, t\rangle\rangle\rangle$
$\lambda FLxy.L(y, F(x))$ $\langle\langle e, e\rangle, \langle\langle e, \langle e, t\rangle\rangle, \langle e, \langle e, t\rangle\rangle\rangle\rangle$
wife: $\lambda x.wife(x)$ $\langle e, e\rangle$
and: $\lambda Pxy.(P(x)\&P(y))$ $\langle\langle e, t\rangle, \langle e, \langle e, t\rangle\rangle\rangle$
bill: $bill$ e
does_too: $\lambda xP.P(x)$ $\langle e, \langle\langle e, t\rangle, t\rangle\rangle$

The types:
F, wife: $\langle e, e\rangle$,
P: $\langle e, t\rangle$,
L, loves: $\langle e, \langle e, t\rangle\rangle$

The grammar:

```
S -> NP VP | SCon S
SCon -> NP VPCon
PP -> 'with' NP
NP -> 'every' 'girl' | 'a' 'book' | 'a' 'boy' | 'his' 'wife'
NP -> NP PP
VPCon -> 'smokes' CON | VP CON
VP -> 'hits' NP | 'loves' NP | 'does' 'too' | 'does_too'
VP -> VP PP | DVP NP | NP VP
DVP -> 'gives' NP
CON -> 'and'
NP -> 'mary' | 'john' | 'bill'
```


S
(loves(john,wife(john)) & loves(bill,wife(bill)))
t
SCon
\y.(loves(john,wife(john)) & loves(y,wife(y)))
<e,t>
S
\P.P(bill)
<<e,t>,t>
NP
john
e
VPCon
\x y.(loves(x,wife(x)) & loves(y,wife(y)))
<e,<e,t>>
NP
bill
e
VP
\x P.P(x)
<e,<<e,t>,t>>
john
john
e
VP
\x.loves(x,wife(x))
<e,t>
CON
\P x y.(P(x) & P(y))
<<e,t>,<e,<e,t>>>
bill
bill
e
does_too
\x P.P(x)
<e,<<e,t>,t>>
loves
\x y.loves(x,y)
<e,<e,t>>
NP
\L x.L(x,wife(x))
<<e,<e,t>>,<e,t>>
and
\P x y.(P(x) & P(y))
<<e,t>,<e,<e,t>>>
his
\F L x.L(x,F(x))
<<e,e>,<<e,<e,t>>,<e,t>>>
wife
\x.wife(x)
<e,e>


S
(loves(john,wife(john)) & loves(bill,wife(john)))
t
SCon
\y.(loves(john,wife(john)) & loves(y,wife(john)))
<e,t>
S
\P.P(bill)
<<e,t>,t>
NP
john
e
VPCon
\x y.(loves(x,wife(john)) & loves(y,wife(john)))
<e,<e,t>>
NP
bill
e
VP
\x P.P(x)
<e,<<e,t>,t>>
john
john
e
VP
\y.loves(y,wife(john))
<e,t>
CON
\P x y.(P(x) & P(y))
<<e,t>,<e,<e,t>>>
bill
bill
e
does_too
\x P.P(x)
<e,<<e,t>,t>>
NP
john
e
VP
\x y.loves(y,wife(x))
<e,<e,t>>
and
\P x y.(P(x) & P(y))
<<e,t>,<e,<e,t>>>
john
john
e
loves
\x y.loves(x,y)
<e,<e,t>>
NP
\L x y.L(y,wife(x))
<<e,<e,t>>,<e,<e,t>>>
his
\F L x y.L(y,F(x))
<<e,e>,<<e,<e,t>>,<e,<e,t>>>>
wife
\x.wife(x)
<e,e>

Exercise 5

conservatity	extension	quantity	example
yes	yes	yes	every
no	no	no	$(E - A) \subset (B \cap Prophets)$, ie., less non-A than B prophets
yes	no	no	$\|(E - A) \cap (B \cap Prophets)\| = 3$, ie., exactly three non-As are B prophets
no	yes	no	$A = (B \cap Prophets)$, ie., all and only A are B prophets
no	no	yes	$(E - A) = B$, ie., all and only non-As are B
yes	yes	no	$(A \cap John) \subset B$, ie., all John's A are B
yes	no	yes	$A \cup B = E$, ie., everyone is A or B
no	yes	yes	$A = B$, ie., all and only As are Bs

Proofs for the first two examples:

- Conservativity of ”every”; this determiner passes the canonical conservativity test:

  every man smiles $\Leftrightarrow$ every man is a man that smiles
- Extension of “every”: if we imagine two domains (eg., corresponding to a context of utterance), one a subset of the other, while the set of men and that of smiling people remains the same, then ”every man smiles” is equivalent in both domains, which is the condition for extension.
- Quantity of “every”: permuting elements of the domain, if done systematically, does not change the meaning of “every”, because it depends only on the elements of A and B, not on any other non-permuted predicate, hence it has quantity.

- Conservativity of “there are less non-As than B prophets”; this quantifier obviously fails the conservativity test:

  There are less non-men than smiling prophets $\not\Leftrightarrow$ There are less non-men than non-men that are smiling prophets
- Extension of “there are less non-As than B prophets”: the definition clearly references the domain E by referring to ”non-As”, so it is likely to fail the extension test. To see this, imagine one domain with two smilers, one man and one woman; a second domain also has two smilers and one man, but two women; everyone is a prophet in both domains. Now, in the first domain the quantifier is true, yet in the second it is false, which proves that it is context-dependent.
- Quantity of “there are less non-A than B prophets”: Imagine a *deus ex machina* wreaking havoc by permuting the set of smiling people.

  men = { Moses, Abraham }
  prophet = { Abraham, Moses }
  smilers = { Abraham, Sarah }
  women = { Sarah }

  Clearly the quantifier is false before the permutation, because there is one non-man but also one smiling prophet. However, after permuting the domain of the smilers and the men (yet crucially not that of the prophets) the quantifier is true:

  men = { Sarah, Abraham }
  prophet = { Abraham, Moses }
  smilers = { Abraham, Moses }
  women = { Moses }

Exercise 6