6.1.xa. Exercise answers

1. **a.** *Ann introduced Bill to Carol*
[λxyz (x *introduced* y *to* z)] *Ann Bill Carol*

Iabc

I `fits` a, b, `'n` c

[I: λxyz (x *introduced* y *to* z); a: *Ann*; b: *Bill*; c: *Carol*]

b. *Ann gave the book to either Bill or Carol*
Ann gave the book to Bill ∨ *Ann gave the book to Carol*
[λxyz (x *gave* y *to* z)] *Ann the book Bill* ∨ [λxyz (x *gave* y *to* z)] *Ann the book Carol*

Gakb ∨ Gakc

`either` G `fits` a, k, `'n` b `or` G `fits` a, k, `'n` c

[G: λxyz (x *gave* y *to* z); a: *Ann*; b: *Bill*; c: *Carol*; k: *the book*]

c. *Ann gave the book to Bill and he gave it to Carol*
Ann gave the book to Bill ∧ *Bill gave the book to Carol*
[λxyz (x *gave* y *to* z)] *Ann the book Bill* ∧ [λxyz (x *gave* y *to* z)] *Bill the book Carol*

Gakb ∧ Gbkc

`both` G `fits` a, k, `'n` b `and` G `fits` b, k, `'n` c

[G: λxyz (x *gave* y *to* z); a: *Ann*; b: *Bill*; c: *Carol*; k: *the book*]

d. *Tom had the package sent to Sue, but it was returned to him*
Tom had the package sent to Sue ∧ *the package was returned to Tom*
[λxyz (x *had* y *sent to* z)] *Tom the package Sue* ∧ [λxy (x *was returned to* y)] *the package Tom*

Htps ∧ Rpt

`both` H `fits` t, p, `'n` s `and` R `fits` p `'n` t

[H: λxyz (x *had* y *sent to* z); R: λxy (x *was returned to* y); p: *the package*; s: *Sue*; t: *Tom*]

e. *Georgia will see Ed if she gets to Denver before Saturday*
Georgia will see Ed ← *Georgia will get to Denver before Saturday*
[λxy (x *will see* y)] *Georgia Ed* ← [λxyz (x *will get to* y *before* z)] *Georgia Denver Saturday*

Sge ← Ggds

Ggds → Sge

`if` G `fits` g, d, `'n` s `then` S `fits` g `'n` e

[G: λxyz (x *will get to* y *before* z); S: λxy (x *will see* y); d: *Denver*; e: *Ed*; g: *Georgia*; s: *Saturday*]

f. *If the murderer is either the butler or the nephew, then I'm Sherlock Holmes*

the murderer is either the butler or the nephew → *I'm Sherlock Holmes*

(*the murderer is the butler* ∨ *the murderer is the nephew*) → *I* = *Sherlock Holmes*

(*the murderer* = *the butler* ∨ *the murderer* = *the nephew*) → i = s

(m = b ∨ m = n) → i = s

`if either` m `is` b `or` m `is` n `then` i `is` s

[b: *the butler*; i: *I*; m: *the murderer*; n: *the nephew*; s: *Sherlock Holmes*]

g. *Neither Ann nor Bill saw Tom speak to either Mike or Nancy*

¬ (*Ann saw Tom speak to either Mike or Nancy* ∨ *Bill saw Tom speak to either Mike or Nancy*)

¬ ((*Ann saw Tom speak to Mike* ∨ *Ann saw Tom speak to Nancy*) ∨ (*Bill saw Tom speak to Mike* ∨ *Bill saw Tom speak to Nancy*))

¬ (([λxyz (x *saw* y *speak to* z)] *Ann Tom Mike* ∨ [λxyz (x *saw* y *speak to* z)] *Ann Tom Nancy*) ∨ ([λxyz (x *saw* y *speak to* z)] *Bill Tom Mike* ∨ [λxyz (x *saw* y *speak to* z)] *Bill Tom Nancy*))

¬ ((Satm ∨ Satn) ∨ (Sbtm ∨ Sbtn))

`not either either` S `fits` a, t, `'n` m `or` S `fits` a,t, `'n` n `or either` S `fits` b,t, `'n` m `or` S `fits` b,t, `'n` n

[S: λxyz (x *saw* y *speak to* z); a: *Ann*; b: *Bill*; m: *Mike*; n: *Nancy*; t: *Tom*]

h. *Tom will agree if each of Ann, Bill, and Carol asks him*

Tom will agree ← *each of Ann, Bill, and Carol will ask Tom*

Tom will agree ← ((*Ann will ask Tom* ∧ *Bill will ask Tom*) ∧ *Carol will ask Tom*)

[λx (x *will agree*)] *Tom* ← (([λxy (x *will ask* y)] *Ann Tom*

∧ [λxy (x *will ask* y)] *Bill* *Tom*) ∧ [λxy (x *will ask* y)] *Carol* *Tom*)

Gt ← ((Aat ∧ Abt) ∧ Act)
((Aat ∧ Abt) ∧ Act) → Gt
`if both both` A `fits` a `'n` t `and` A `fits` b `'n` t `and` A `fits` c `'n` t `then` G `fits` t

[A: λxy (x *will ask* y); G: λx (x *will agree*); a: *Ann*; b: *Bill*; c: *Carol*; t: *Tom*]

The function of *each* here is to indicate a group of two-place predication rather than a single four-place predicate λxyzw (x, y, *and* z *will ask* w), which is what would be required in order to express instead the idea of Ann, Bill, and Carol making the request as a group.

2. **a.** [λxy (x *is west of* y)] *Crawfordsville* *Indianapolis* ∧ [λxy (x *is south of* y)] *Crawfordsville* *Lafayette*
Crawfordsville is west of Indianapolis ∧ *Crawfordsville is south of Lafayette*
Crawfordsville is west of Indianapolis and south of Lafayette

b. [λxy (x *has met* y)] *Ann* *Bill* → [λxy (x *has met* y)] *Bill* *Ann*
Ann has met Bill → *Bill has met Ann*
If Ann has met Bill then he has met her

c. [λxyz (x *introduced* y *to* z)] *Alice* *Clarice* *Boris* ∧ [λxyz (x *introduced* y *to* z)] *Alice* *Doris* *Boris*
Alice introduced Clarice to Boris ∧ *Alice introduced Doris to Boris*
Alice introduced Clarice and Doris to Boris

d. [λxy (x *wrote to* y)] *Alice* *Boris* ∧ [λxyzw (x *asked* y *to write* z *about* w)] *Alice* *Boris* *Alice* *Boris*
Alice wrote to Boris ∧ *Alice asked Boris to write Alice about Boris*
Alice wrote to Boris ∧ *Alice asked Boris to write her about himself*
Alice wrote to Boris and asked him to write her about himself

e. g = c → (f = s ∧ p = t)
Green Bay = *the city* → (*football* = *the sport* ∧ *the Packers*

= *the team*)

Green Bay is the city → (*football is the sport* ∧ *the Packers are the team*)

Green Bay is the city → *football is the sport and the Packers are the team*

If Green Bay is the city, then football is the sport and the Packers are the team

Glen Helman 25 Aug 2005