The tasks assigned below are far from trivial. You may find it helpful to consult the paper,

Champollion, L. (2015) The interaction of compositional semantics and event semantics. *Linguistics and Philosophy, 36:31–66.*

Here is a lexicon that could be an answer to Q2-a of Assignment 4 – remember that v is the type of eventualities.

Lexicon 0.1

$$
\begin{array}{rrr}
\text{walks} := & \lambda x \lambda e.walking' e \wedge agent' e\, x :: & e(vt) \\
\\
\text{killed} := & \lambda x \lambda y \lambda e.killing' e \wedge patient' e\, x \wedge agent' e\, y :: & e(e(vt)) \\
\text{loves} := & \lambda x \lambda y \lambda e.loving' e \wedge theme' e\, x \wedge agent' e\, y :: & e(e(vt)) \\
\text{wrote} := & \lambda x \lambda y \lambda e.writing' e \wedge theme' e\, x \wedge agent' e\, y :: & e(e(vt)) \\
\text{reads} := & \lambda x \lambda y \lambda e.reading' e \wedge theme' e\, x \wedge agent' e\, y :: & e(e(vt)) \\
\\
\text{John} := & \lambda p.p\, j' :: & ett \\
\text{Mary} := & \lambda p.p\, m' :: & ett \\
\\
\text{woman} := & \lambda x.woman' x :: & et \\
\text{book} := & \lambda x.book' x :: & et \\
\text{knife} := & \lambda x.knife' x :: & et \\
\text{letter} := & \lambda x.letter' x :: & et \\
\text{pen} := & \lambda x.pen' x :: & et \\
\\
\text{blue} := & \lambda p \lambda x.blue' x \wedge p\, x :: & et(et) \\
\\
\text{is} := & \lambda p \lambda x.p(\lambda x.x = x) x :: & et(et)(et) \\
\\
\text{no} := & \lambda p \lambda q.\neg(\exists x.px \wedge qx) :: & et(ett) \\
\text{a} := & \lambda p \lambda q.\exists x.px \wedge qx :: & et(ett) \\
\text{every} := & \lambda p \lambda q.\forall x.px \rightarrow qx :: & et(ett) \\
\\
\text{ACC} := & \lambda k \lambda q \lambda y.k(\lambda x.q\, x\, y) :: & ett(e(et)(et)) \\
\text{NOM} := & \lambda p.p :: & ett(ett) \\
\\
\text{with} := & \lambda q \lambda f \lambda x \lambda e.f\, x\, e \wedge q(instr' e) :: & ett(e(vt))(e(vt))
\end{array}
$$

Q2-b could be handled by a lexicon which overwrites the following categories in the previous lexicon:

Lexicon 0.2

$$
\begin{array}{rrr}
\text{walks} := & \lambda e.walking' e :: & vt \\
\\
\text{killed} := & \lambda e.killing' e :: & vt \\
\text{loves} := & \lambda e.loving' e :: & vt \\
\text{reads} := & \lambda e.reading' e :: & vt \\
\\
\text{ACC} := & \lambda k \lambda q \lambda e.qe \wedge k(patient' e) :: & ett(vt(vt)) \\
\text{NOM} := & \lambda k \lambda q \lambda e.qe \wedge k(agent' e) :: & ett(vt(vt)) \\
\\
\text{with} := & \lambda q \lambda f \lambda e.f\, e \wedge q(instr' e) :: & ett(vt(vt))
\end{array}
$$

Both solutions succesfully interpret:

(1) ((NOM John) ((killed (ACC Mary)) (with (a knife)))).

as,

(2) $\lambda e.killing'e \wedge patient'e\,mary' \wedge agent'e\,john' \wedge \exists x.knife'x \wedge instr'e\,x$

This still is not a t type result, it denotes a set of eventualities. It is rather stratightforward to turn this into a t type interpretation with existential closure; and this job can be assigned to an abstract assertion operator that applies after everything gets combined. A suitable ortographic carrier of such function would be the period that ends a sentence – a sentence ending with a question mark would not assert its content. Here is the lexcial entry:

Lexicon 0.3

$$. := \quad \lambda f \exists e.f\,e :: \quad vt(t)$$

With this in hand, we get:

(3) a. ((NOM John) ((killed (ACC Mary)) (with (a knife)))).
b. $\exists e.killing'e \wedge patient'e\,mary' \wedge agent'e\,john' \wedge \exists x.knife'x \wedge instr'e\,x$

which is quite satisfactory.

Q 1. (30%)
The above solutions are not, however, successful with quantified expressions.

(4) a. NOM John wrote ACC a letter with a pen.
b. $\exists e.\exists x.letter'x \wedge \lambda e.writing'e \wedge patient'e\,x \wedge agent'e\,john'e \wedge \exists y.pen'y \wedge instr'e\,y$

(5) a. NOM every man wrote ACC a letter.
b. $\exists e.\forall x.man'x \rightarrow \exists y.letter'y \wedge \lambda e.writing'e \wedge patient'e\,y \wedge agent'e\,x\,e$

The obvious problem with these interpretations is the type mismatches caused by the event type lambda binders. Fix them, so that we get:

(6) a. NOM Every woman wrote ACC every letter with a pen.
b. $\exists e.\forall x.woman'x \rightarrow \forall y.letter'y \rightarrow writing'e \wedge patient'ey \wedge agent'ex \wedge \exists z.pen'z \wedge instr'e\,z$

Build over the lexicon where roles like *agent* and *patient* are contributed by the verb rather than the case markers.

Solution: Here is the full lexicon for the solution (only one examplary item is given for each type of expression):

Lexicon 0.4

walks :=	$\lambda x \lambda e.walking' e \wedge agent' e\, x$ ::	$e(vt)$
wrote :=	$\lambda x \lambda y \lambda e.writing' e \wedge theme' e\, x \wedge agent' e\, y$ ::	$e(e(vt))$
John :=	$\lambda p.p\, j'$ ::	ett
letter :=	$\lambda x.letter' x$ ::	et
blue :=	$\lambda p \lambda x.blue' x \wedge p\, x$ ::	$et(et)$
is :=	$\lambda p \lambda x.p(\lambda x.x = x)x$ ::	$et(et)(et)$
no :=	$\lambda p \lambda q.\neg(\exists x.px \wedge qx)$ ::	$et(ett)$
a :=	$\lambda p \lambda q.\exists x.px \wedge qx$ ::	$et(ett)$
every :=	$\lambda p \lambda q.\forall x.px \rightarrow qx$ ::	$et(ett)$
ACC :=	$\lambda k \lambda q \lambda y \lambda e.k(\lambda x.q\, x\, y\, e)$ ::	$ett(e(e(vt))(e(vt)))$
NOM :=	$\lambda f \lambda g \lambda e.f(\lambda x.g\, x\, e)$ ::	$ett(e(vt)(vt))$
with :=	$\lambda q \lambda f \lambda x \lambda e.f\, x\, e \wedge q(instr' e)$ ::	$ett(e(vt))(e(vt))$
. :=	$\lambda f \exists e.f\, e$ ::	$vt(t)$

Q 2. (50%)

There remains a problem. The interpretation (6b) depicts a single writing event that every woman in the model contributes to. But the sentence (6a), at least in its most prominent reading, is true in a situation where there are as many letter writing events as there are women in the model. Therefore a more accurate interpretation would be:

(7) a. NOM every man wrote ACC a letter.
 b. $\forall x.woman' x \rightarrow \exists y.letter' y \wedge \exists e.writing' e \wedge patient' e\, y \wedge agent' e\, x$

You will need to modify your solution to the previous question to be able to get this type of readings. One way to do it is to give verbs the following definitions:

$$\text{wrote} := \lambda x \lambda y \lambda f.\exists e.writing' e \wedge patient' e\, x \wedge agent' e\, y \wedge f\, e :: \quad e(e(vtt))$$
$$\text{walks} := \lambda x \lambda f.\exists e.walking' e \wedge agent' e\, x \wedge f\, e :: \quad e(vtt)$$

In this case your assertion operator that closes the formula would be some function that gets rid of f:

$$. := \lambda s.s(\lambda e.true') :: \quad vttt$$

where $true'$ is a constant that always evaluates to 1 in the model; therefore any part $\wedge true'$ can safely be deleted from formulas.

With appropriate adjustments to other items you should be able to arrive at the following interpretations:

(8) a. (((NOM John) (wrote (ACC (every letter)))) .)
 b. $\forall x.letter' x \rightarrow \exists e.writing' e \wedge patient' e\, x \wedge agent' e\, john' \wedge true'$

(9) a. NOM every woman wrote ACC every letter .
 b. $\forall x.woman' x \rightarrow \forall y.letter' y \rightarrow \exists e.writing' e \wedge patient' ey \wedge agent' ex \wedge true'$

Solution:

Lexicon 0.5

walks :=	$\lambda x \lambda f.\exists e.walking' e \wedge agent' e\, x \wedge f\, e$ ::	$e(vtt)$
wrote :=	$\lambda x \lambda y \lambda f.\exists e.writing' e \wedge patient' e\, x \wedge agent' e\, y \wedge f\, e$ ::	$e(e(vtt))$
John :=	$\lambda p.p\, j'$ ::	ett
letter :=	$\lambda x.letter' x$ ::	et
blue :=	$\lambda p \lambda x.blue' x \wedge p\, x$ ::	$et(et)$
is :=	$\lambda p \lambda x.p(\lambda x.x = x)x$ ::	$et(et)(et)$
no :=	$\lambda p \lambda q.\neg(\exists x.px \wedge qx)$ ::	$et(ett)$
a :=	$\lambda p \lambda q.\exists x.px \wedge qx$ ::	$et(ett)$
every :=	$\lambda p \lambda q.\forall x.px \rightarrow qx$ ::	$et(ett)$
ACC :=	$\lambda k \lambda p \lambda y \lambda f.k(\lambda x.pxyf)$ ::	$ett(e(e(vt))(e(vt)))$
NOM :=	$\lambda k \lambda q \lambda f.k(\lambda x.qxf)$ ::	$ett(e(vt)(vt))$
with :=	$\lambda q \lambda f \lambda x \lambda e.f\, x\, e \wedge q(instr' e)$ ::	$ett(e(vt))(e(vt))$
. :=	$\lambda s.s(\lambda e.true')$ ::	$vttt$

Q 3. (20%)

Once the above issues are fixed, adverbial modification can be taken care of. Your lexicon should be able to derive the following:

(10) a. NOM every woman wrote ACC every letter with a pen .
b. $\forall x.woman' x \rightarrow \exists y.pen' y \wedge \forall z.letter' z \rightarrow \exists e.writing' e \wedge patient' e\, z \wedge agent' e\, x \wedge instr' e\, y \wedge true'$

(11) a. NOM every woman wrote ACC every letter at a desk with a blue pen .
b. $\forall x.woman' x \rightarrow \exists y.pen' y \wedge blue' y \wedge \exists z.desk' z \wedge \forall s.letter' s \rightarrow \exists e.writing' e \wedge patient' e\, s \wedge agent' e\, x \wedge loc' e\, z \wedge instr' e\, y \wedge$

and, so on. . .

Solution: You need a new definition for the adverbial introducing prepositions like *with* or *at*:

with :=	$\lambda q \lambda k \lambda x \lambda f.q(\lambda z.k\, x(\lambda e.instr' e\, z \wedge f\, e))$ ::	$ett(e(vtt)(e(vtt)))$
at :=	$\lambda q \lambda k \lambda x \lambda f.q(\lambda z.k\, x(\lambda e.loc' e\, z \wedge f\, e))$ ::	$ett(e(vtt)(e(vtt)))$