The relational algebra is a small language for manipulating sets of tuples, and is one of the central objects of study in database theory. We can give a syntax for (a subset of) it as follows:

\[
\begin{align*}
\tau & ::= \text{int} \mid \text{bool} & \text{Data types} \\
d & ::= n \mid b & \text{Data values} \\
R & ::= [l_1 : \tau_1, \ldots, l_n : \tau_n] & \text{Record types (with disjoint field names } l_i) \\
r & ::= [l_1 = v_1, \ldots, l_n = v_n] & \text{Record values (with disjoint field names } l_i) \\
S & ::= \text{Set } R & \text{Set types} \\
e & ::= \{r_1, \ldots, r_n\} & \text{Set literal} \\
\mid e \cup e' & & \text{Set union} \\
\mid e \times e' & & \text{Cartesian product with disjoint field labels} \\
\mid \Pi_{l_1,\ldots,l_n}(e) & & \text{Records of } e \text{ with fields not in } l_1,\ldots,l_n \text{ removed} \\
\mid \sigma_{l_1=l_2}(e) & & \text{Subset of } e \text{ where the fields } l_1 \text{ and } l_2 \text{ are equal}
\end{align*}
\]

(a) State the form of the typing judgements for this language, and give typing rules for this programming language ascribing to each category of terms its corresponding types. \[8 \text{ marks}\]

(b) Define a deterministic small-step operational semantics for this language, defining any auxiliary functions you need as well. \[10 \text{ marks}\]

(c) Give a precise statement of the progress and preservation properties for this language. You do not need to give a proof. \[2 \text{ marks}\]