Exercise 2.2

In this exercise we work in a language which contains a single binary predicate symbol E and two constant symbols c and d. The structures of this language are (finite or infinite) graphs with two designated vertices, a source c and a sink d. A path is a finite list of vertices connected by edges. The length of a path is the number of edges it contains. Show that:

1. For every $k \in \mathbb{N}$ the graphs containing a path of length k from c to d are definable by a first-order sentence.
2. The graphs which do contain a path from c to d,
   (a) are not first-order definable.
   (b) are definable by a second-order sentence.
3. The graphs which do not contain a path from c to d
   (a) are not definable by a first-order sentence.
   (b) are first-order definable.
   (c) are definable by a second-order sentence.

We fix $I(c) = c^{\mathcal{I}}$ and $I(d) = d^{\mathcal{I}}$.

1. For every $k \in \mathbb{N}$ the graphs containing a path of length k from c to d are definable by a first-order sentence.

   Consider the following sentence.

   $$C_k := \exists x_1 \dots \exists x_{k-1} \left(E(c, x_1) \wedge \bigwedge_{i=1}^{k-2} E(x_i, x_{i+1}) \wedge E(x_{k-1}, d) \right)$$

   Clearly, this sentence can only be true if there exists at least one list of length k of the form $E^{\mathcal{I}}(c^{\mathcal{I}}, x_1^{\mathcal{I}}), \dots E^{\mathcal{I}}(x_{k-1}^{\mathcal{I}}, d^{\mathcal{I}})$ where the end point of the prior edge is the starting point of the latter, which is precisely the definition of a path.

2. The graphs which do contain a path from c to d,

   (a) are not first-order definable.

   Assume that this is the case. Hence, there exists a set of first order sentences C (in this language) that is true if and only if there exists a path from $c^{\mathcal{I}}$ to $d^{\mathcal{I}}$. Let $D_k := \{\neg C_i \forall 0 \leq i \leq k\}$.

   $$C_\omega := D \cup P = \{\neg C_k \mid \forall k \geq 0\} \cup C$$

Consider the finite subset $X_k \underset{fin.}{\subset} C_\omega$ of the form $X_k = D' \cup C'$ with $C' \underset{fin.}{\subset} C$ and $D' \underset{fin.}{\subset} D$. Clearly, since X_k is finite, there exists a maximal k such that $D' \subseteq D_k$. Thus any model with a path greater k can satisfy D'. That is, consider the following interpretation $\mathcal{G} \coloneqq \langle V, I \rangle$ such that $V = \{v_i \mid 0 \leq i \leq k+1\}$ such that $c^{\mathcal{I}} = v_0$, $d^{\mathcal{I}} = (v_{k+1})$ and $I(E) \coloneqq \{(v_i, v_{i+1}) \mid 0 \leq i \leq k\}$. Since there exists a path in $\mathcal{G}$ it follows by assumption that $\mathcal{G} \vDash C$ and thus especially $\mathcal{G} \vDash C'$. Hence, $\mathcal{G}$ satisfies X_k. Since this can be done for arbitrary k, one can conclude that any finite subset of C_ω is satisfiable. Hence, by compactness one obtains that there exists a model $\mathcal{G}_\omega$ that satisfies C_ω. Clearly, it is impossible that there exists a finite path from $c^{\mathcal{I}}$ to $d^{\mathcal{I}}$ in $\mathcal{G}_\omega$, i.e. otherwise there exists a k such that a C_k is violated thus D would not be satisfied by the structure. However, since $\mathcal{G}_\omega \vDash C$, by assumption, there must exists a path from $c^{\mathcal{I}}$ to $d^{\mathcal{I}}$. Hence, one arrives at a contradiction.

(b) are definable by a second-order sentence.

The intent is to formalise reachability, with respect to $E^{\mathcal{I}}$. That is, there exists a path from $c^{\mathcal{I}}$ to $d^{\mathcal{I}}$, if and only if, every set that contains the transitive closure of $E^{\mathcal{I}}$ contains the element $(c^{\mathcal{I}}, d^{\mathcal{I}})$. That is, consider the following formulas

$$R_{\subseteq}(X, Y) \coloneqq \forall x \forall y (X(x, y) \rightarrow Y(x, y))$$
$$R_{\rightarrow}(X) \coloneqq \forall x \forall y \forall z (X(x, y) \rightarrow X(y, z) \rightarrow X(x, z))$$

The first formula forces that $X^{\mathcal{I}}$ is a sub-relation of $Y^{\mathcal{I}}$, i.e. if $(x, y) \in X^{\mathcal{I}}$ then it must be the case that $(x, y) \in Y^{\mathcal{I}}$ and therefore $X^{\mathcal{I}} \subseteq Y^{\mathcal{I}}$. The second formula requires the input binary relation $X^{\mathcal{I}}$ to be transitive. Using those two one can now construct the following sentence.

$$C \coloneqq \forall X (R_{\subseteq}(E, Y) \rightarrow R_{\rightarrow}(X) \rightarrow R(c, d))$$

That is, C requires that if an relation $R^{\mathcal{I}}$ that contains $E^{\mathcal{I}} \subseteq R^{\mathcal{I}}$ is also transitive closed then it must contain $(c^{\mathcal{I}}, d^{\mathcal{I}})$. What remains to show is that this formula forces the required behaviour.

Consider a structure $\mathcal{G}$, show that $\mathcal{G} \vDash C$ if and only if there exists a path in $\mathcal{G}$ from c to d. Assume that there exists a path from c^I to d^I. Then it must be the case that (c^I, d^I) is in the transitive closure of $E^{\mathcal{I}}$, i.e. $E^{\mathcal{I}^*}$. Now, take an relation $R^{\mathcal{I}} \subset D^2$ such that $E^{\mathcal{I}} \subseteq R^{\mathcal{I}}$ and that $R^{\mathcal{I}}$ is closed transitively. Clearly, $E^{\mathcal{I}^*} \subseteq R^{\mathcal{I}}$

and therefore $(c^I, d^I) \in R^{\mathcal{I}}$. On the other hand, assume that there does not exists a path from c^I to d^I. Then it must be the case that $(c^I, d^I) \notin I(E)^*$. However, due to the fact that $E^{\mathcal{I}} \subseteq E^{\mathcal{I}^*}$ and that $E^{\mathcal{I}^*}$ is closed transitively by definition, it follows that one has found a relation that invalidates the sentence C.

3. The graphs which do not contain a path from c to d

   (a) are not definable by a first-order sentence.

   Assume there exists such a sentence. Let this sentence be called S. Since S expresses that there does not exists a path from c to d it follows that its negation $\neg S$ expresses that there does exists a path from $c^{\mathcal{I}}$ to $d^{\mathcal{I}}$. However, consider the following set of sentences $\{\neg S\}$. Clearly, this set of sentences can only be satisfied if there exists a path from $c^{\mathcal{I}}$ to $d^{\mathcal{I}}$ in the respective structure. However, this would imply that the graphs which do contain a path from $c^{\mathcal{I}}$ to $d^{\mathcal{I}}$, are first-order definable. Which clearly contradicts the statement made above.

   (b) are first-order definable.

   Recall that the sentence

   $$C_k := \exists x_1 \ldots \exists x_{k-1} \left(E(c, x_1) \wedge \bigwedge_{i=1}^{k-2} E(x_i, x_{i+1}) \wedge E(x_{k-1}, d) \right)$$

   expresses that there exists a path of length k from $c^{\mathcal{I}}$ to $d^{\mathcal{I}}$. Hence, its negation

   $$\neg C_k = \forall x_1 \ldots \forall x_{k-1} \left(\neg E(c, x_1) \vee \bigvee_{i=1}^{k-2} \neg E(x_i, x_{i+1}) \vee \neg E(x_{k-1}, d) \right)$$

   expresses that no path of length k exits from $c^{\mathcal{I}}$ to $d^{\mathcal{I}}$. Consider the following set of sentences

   $$D_k := \{\neg C_i \mid \forall 0 \leq i \leq k\}$$

   A structure $\mathcal{G}$ such that $\mathcal{G} \vDash D_k$, can not have a path of length $1, 2, \ldots k$ from $c^{\mathcal{I}}$ to $d^{\mathcal{I}}$. Therefore a structure $\mathcal{G}'$ that satisfies

   $$D := \{D_k \mid \forall k \geq 0\}$$

   can not have a path of any finite length k from $c^{\mathcal{I}}$ to $d^{\mathcal{I}}$. That is, for any k there exists a $D_k \subseteq D$ that prevents due to $\mathcal{G}' \vDash D_k$ the existence of a path smaller that k (from $c^{\mathcal{I}}$ to $d^{\mathcal{I}}$). However, since a path has to be finite it follows that if a structure satisfies D it can not have a path from $c^{\mathcal{I}}$ to $d^{\mathcal{I}}$.

(c) are definable by a second-order sentence. In a previous exercise it was already shown that the graphs which do contain a path from c to d, are definable by a second-order sentence. Hence, its negation can only be true if the satisfying structure does not contain a path from c to d. That is,

$$\neg C = \exists X(R_{\subseteq}(E, Y) \wedge R_{\rightarrow}(X) \wedge \neg R(c, d))$$

Exercise 3.2

We have defined $\bot$ and $\neg$ as simply typed expressions. Analogously to these, define binary disjunction $\vee$, binary conjunction $\wedge$, logical equivalence $\leftrightarrow$ and the existential quantifier $\exists_\tau$ as simply typed expression. Give type derivations and show that the semantics has the expected behaviour.

Note: To increase readability, there are references to previously made statements. That is, if in the derivation $\neg$ is used, only an reference to the original derivation of the operator will be given. Similarly, $\wedge$ will be referenced by a derivation that occurs after the derivation of $\wedge$. Moreover, the same holds for the demonstration of the desired semantic behaviour. That is, once it was show that the term for $\wedge$ has the desired semantics, its semantics will be used directly in future demonstrations.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(\textit{see script})}{\lambda x^o.\, \supset\, x^o\, \bot : o \to o}}{\neg : o \to o} \qquad x^o : o}}{\supset : o \to o \to o} \qquad \neg x^o : o}}{\supset\, \neg x^o : o \to o} \qquad y^o : o}}{\supset\, (\neg x^o)\, y^o : o}}{\lambda y^o.\, \supset\, (\neg x^o)\, y^o : o \to o}}{\lambda x^o \lambda y^o.\, \supset\, (\neg x^o)\, y^o : o \to o \to o}}{\vee : o \to o \to o}$$

$$\begin{aligned}
I(\vee)(a, b) &= (I \cup \{x^o \mapsto a, y^o \mapsto b\})(\supset\ (\neg x^o)\ y^o) \\
&= I'(\supset\ (\neg x^o)\ y^o) \\
&= I'(\supset)(I'(\neg x^o), I'(y^o)) \\
&= \begin{cases} \text{true} & \textit{if } I'(\neg x^o) = \text{false } \textit{or } I'(y^o) = \text{true} \\ \text{false} & \textit{otw.} \end{cases} \\
&= \begin{cases} \text{true} & \textit{if } I'(x^o) = \text{true } \textit{or } I'(y^o) = \text{true} \\ \text{false} & \textit{otw.} \end{cases}
\end{aligned}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\supset : o \to o \to o}\quad \dfrac{}{x^o : o}}{\supset\ x^o : o \to o}\quad \dfrac{\dfrac{\dfrac{\dfrac{}{\lambda x^o.\ \supset\ x^o\ \bot : o \to o}\ (\textit{see script})}{\neg : o \to o}\quad \dfrac{}{y^o : o}}{\neg y^o : o}}{\supset\ x^o\ (\neg y^o) : o \to o}}{\neg : o \to o}}{\neg\ (\supset\ x^o\ (\neg y^o)) : o}}{\lambda y^o.\neg\ (\supset\ x^o\ (\neg y^o)) : o \to o}}{\lambda x^o \lambda y^o.\neg\ (\supset\ x^o\ (\neg y^o)) : o \to o \to o}$$
$$\wedge : o \to o \to o$$

$$\begin{aligned}
I(\wedge)(a,b) &= (I \cup \{x^o \mapsto a, y^o \mapsto b\})(\neg\ (\supset\ x^o\ (\neg y^o))) \\
&= \begin{cases} \text{true} & \textit{if } I'(\supset\ x^o\ (\neg y^o)) = \text{false} \\ \text{false} & \textit{if } I'(\supset\ x^o\ (\neg y^o)) = \text{true} \end{cases} \\
&= \begin{cases} \text{true} & \textit{if not } (I'(x^o) = \text{false } \textit{and } I'(\neg y^o) = \text{true}) \\ \text{false} & \textit{if } I'(x^o) = \text{false } \textit{or } I'(\neg y^o) = \text{true} \end{cases} \\
&= \begin{cases} \text{true} & \textit{if } I'(x^o) = \text{true } \textit{and } I'(y^o) = \text{true} \\ \text{false} & \textit{if } I'(x^o) = \text{false } \textit{or } I'(y^o) = \text{false} \end{cases} \\
&= \begin{cases} \text{true} & \textit{if } I'(x^o) = \text{true } \textit{and } I'(y^o) = \text{true} \\ \text{false} & \textit{otw.} \end{cases}
\end{aligned}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\wedge : o \to o \to o}\ (\textit{see above})\quad \dfrac{\dfrac{\dfrac{}{\supset^o : o \to o \to o}\quad \dfrac{}{x^o : o}}{\supset\ x^o : o}\quad \dfrac{}{y^o : o}}{\supset\ x^o\ y^o : o}}{\wedge\ (\supset\ x^o\ y^o) : o \to o}\quad \dfrac{\dfrac{\dfrac{}{\supset^o : o \to o \to o}\quad \dfrac{}{y^o : o}}{\supset\ y^o : o}\quad \dfrac{}{x^o : o}}{\supset\ y^o\ x^o : o}}{\wedge\ (\supset\ x^o\ y^o)\ (\supset\ y^o\ x^o) : o}}{\lambda y^o.\wedge\ (\supset\ x^o\ y^o)\ (\supset\ y^o\ x^o) : o \to o}}{\lambda x^o \lambda y^o.\wedge\ (\supset\ x^o\ y^o)\ (\supset\ y^o\ x^o) : o \to o \to o}$$
$$\leftrightarrow : o \to o \to o$$

$$\begin{aligned}
I(\leftrightarrow)(a,b) &= (I \cup \{x^o \mapsto a, y^o \mapsto b\})(\wedge\ (\supset\ x^o\ y^o)\ (\supset\ y^o\ x^o)) \\
&= \begin{cases} \text{true} & \textit{if } I'(\supset\ x^o\ y^o) \textit{ and } I'(\supset\ y^o\ x^o) \\ \text{false} & \textit{otw.} \end{cases} \\
&= \begin{cases} \text{true} & \textit{if } (I'(x^o) = \text{false } \textit{or } I'(y^o) = \text{true}) \textit{ and } (I'(y^o) = \text{false } \textit{or } I'(x^o) = \text{true}) \\ \text{false} & \textit{otw.} \end{cases} \\
&= \begin{cases} \text{true} & \textit{if } (I'(x^o) = \text{false } \textit{and } I'(y^o) = \text{false}) \textit{ or } (I'(x^o) = \text{true } \textit{and } I'(x^o) = \text{true}) \\ \text{false} & \textit{otw.} \end{cases}
\end{aligned}$$

$$\dfrac{\dfrac{\neg : o \to o \qquad \dfrac{\forall_\tau : (\tau \to o) \to o \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\lambda x^o.\, \supset x^o \bot : o \to o}\ (see\ script)}{\neg : o \to o} \qquad x^o : o}{\neg x^o : o}}{\forall_\tau \neg x^{\tau \to o} : o}}{\neg (\forall_\tau \neg x^{\tau \to o}) : o}}{\lambda x^{\tau \to o}. \neg (\forall_\tau \neg x^{\tau \to o}) : (\tau \to o) \to o}}{\exists_\tau : (\tau \to o) \to o}$$

$$\begin{aligned} I(\exists_\tau)(\varphi) &= (I \cup \{x^{\tau \to o} \mapsto \varphi\})(\neg (\forall_\tau x^{\tau \to o})) \\ &= \begin{cases} \text{true} & \textit{if not } (I'(\varphi(m)) = \text{true for all } m \in D_\tau) \\ \text{false} & \textit{otw.} \end{cases} \\ &= \begin{cases} \text{true} & \textit{if } I'(\neg \varphi(m)) = \text{false for some } m \in D_\tau \\ \text{false} & \textit{otw.} \end{cases} \\ &= \begin{cases} \text{true} & \textit{if } I'(\varphi(m)) = \text{true for some } m \in D_\tau \\ \text{false} & \textit{otw.} \end{cases} \end{aligned}$$

Exercise 3.4

Show that the following formulas are provable in $\mathbf{NK}_\omega$ for arbitrary types τ, σ:

1. *reflexivity:* $\forall x^\tau x =_\tau x$
2. *symmetry:* $\forall x^\tau \forall y^\tau (x =_\tau y \supset y =_\tau x)$
3. *transitivity:* $\forall x^\tau \forall y^\tau \forall z^\tau (x =_\tau y \supset y =_\tau z \supset x =_\tau z)$
4. *compatibility:* $\forall f^{\tau \to \sigma} \forall x^\tau \forall y^\tau (x =_\tau y \supset fx =_\sigma fy)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[P^{\tau \to o} a^\tau]^1}{P^{\tau \to o} a^\tau \supset P^{\tau \to o} a^\tau}\,(\supset_I^1) \qquad \dfrac{[P^{\tau \to o} a^\tau]^1}{P^{\tau \to o} a^\tau \supset P^{\tau \to o} a^\tau}\,(\supset_I^1)}{(P^{\tau \to o} a^\tau \supset P^{\tau \to o} a^\tau) \wedge (P^{\tau \to o} a^\tau \supset P^{\tau \to o} a^\tau)}\,(Def.)}{P^{\tau \to o} a^\tau \leftrightarrow P^{\tau \to o} a^\tau}\,(\forall_I)}{\forall Y^{\tau \to o} (Y\ a^\tau \leftrightarrow Y\ a^\tau)}\,(=_{\alpha\beta})}{(\lambda x^\tau \lambda y^\tau . \forall Y^{\tau \to o} (Y\ x \leftrightarrow Y\ y))\ a^\tau\ a^\tau}\,(Def.)}{a^\tau =_\tau a^\tau}\,(\forall_I)}{\forall x^\tau x =_\tau x}$$

Left branch:

$$[a =_\tau b]^1$$
$$\overline{(\lambda x^\tau \lambda y^\tau . \forall Y^{\tau\to o}(Y\ x \leftrightarrow Y\ y))\ a^\tau\ b^\tau}\ (Def.)$$
$$\overline{\forall Y^{\tau\to o}(Y\ a^\tau \leftrightarrow Y\ b^\tau)}\ (=_{\alpha\beta})$$
$$\overline{P^{\tau\to o}a^\tau \leftrightarrow P^{\tau\to o}b^\tau}\ (\forall_E)$$
$$\overline{(P^{\tau\to o}a^\tau \supset P^{\tau\to o}b^\tau) \wedge (P^{\tau\to o}b^\tau \supset P^{\tau\to o}a^\tau)}\ (Def.)$$
$$\overline{P^{\tau\to o}b^\tau \supset P^{\tau\to o}a^\tau}\ (\wedge_E)$$

Right branch:

$$[a =_\tau b]^1$$
$$\overline{(\lambda x^\tau \lambda y^\tau . \forall Y^{\tau\to o}(Y\ x \leftrightarrow Y\ y))\ a^\tau\ b^\tau}\ (Def.)$$
$$\overline{\forall Y^{\tau\to o}(Y\ a^\tau \leftrightarrow Y\ b^\tau)}\ (=_{\alpha\beta})$$
$$\overline{P^{\tau\to o}a^\tau \leftrightarrow P^{\tau\to o}b^\tau}\ (\forall_E)$$
$$\overline{(P^{\tau\to o}a^\tau \supset P^{\tau\to o}b^\tau) \wedge (P^{\tau\to o}b^\tau \supset P^{\tau\to o}a^\tau)}\ (Def.)$$
$$\overline{P^{\tau\to o}a^\tau \supset P^{\tau\to o}b^\tau}\ (\wedge_E)$$

Combined:

$$\overline{(P^{\tau\to o}b^\tau \supset P^{\tau\to o}a^\tau) \wedge (P^{\tau\to o}a^\tau \supset P^{\tau\to o}b^\tau)}\ (\wedge_I)$$
$$\overline{P^{\tau\to o}b^\tau \leftrightarrow P^{\tau\to o}a^\tau}\ (Def.)$$
$$\overline{\forall Y^{\tau\to o}(Y\ b^\tau \leftrightarrow Y\ a^\tau)}\ (\forall_I)$$
$$\overline{(\lambda x^\tau \lambda y^\tau . \forall Y^{\tau\to o}(Y\ x \leftrightarrow Y\ y))\ b^\tau\ a^\tau}\ (=_{\alpha\beta})$$
$$\overline{b^\tau =_\tau a^\tau}\ (Def.)$$
$$\overline{a^\tau =_\tau b^\tau \supset b^\tau =_\tau a^\tau}\ (\supset_I^1)$$
$$\overline{\forall y^\tau (a^\tau =_\tau y \supset y =_\tau a^\tau)}\ (\forall_I)$$
$$\overline{\forall x^\tau \forall y^\tau (x =_\tau y \supset y =_\tau x)}\ (\forall_I)$$

$$\frac{\overset{(a)}{P^{\tau\to o}a^\tau \supset P^{\tau\to o}c^\tau} \qquad \overset{(b)}{P^{\tau\to o}c^\tau \supset P^{\tau\to o}a^\tau}}{(P^{\tau\to o}a^\tau \supset P^{\tau\to o}c^\tau) \wedge (P^{\tau\to o}c^\tau \supset P^{\tau\to o}a^\tau)}\ (\wedge_I)$$
$$\overline{P^{\tau\to o}a^\tau \leftrightarrow P^{\tau\to o}c^\tau}\ (Def.)$$
$$\overline{\forall Y^{\tau\to o}(Y\ a^\tau \leftrightarrow Y\ c^\tau)}\ (\forall_I)$$
$$\overline{(\lambda x^\tau \lambda y^\tau . \forall Y^{\tau\to o}(Y\ x \leftrightarrow Y\ y))\ a^\tau\ c^\tau}\ (=_{\alpha\beta})$$
$$\overline{a^\tau =_\tau c^\tau}\ (Def.)$$
$$\overline{b^\tau =_\tau c^\tau \supset a^\tau =_\tau c^\tau}\ (\supset_I^2)$$
$$\overline{a^\tau =_\tau b^\tau \supset b^\tau =_\tau c^\tau \supset a^\tau =_\tau c^\tau}\ (\supset_I^1)$$
$$\overline{\forall z^\tau (a^\tau =_\tau b^\tau \supset b^\tau =_\tau z \supset a^\tau =_\tau z)}\ (\forall_I)$$
$$\overline{\forall y^\tau \forall z^\tau (a^\tau =_\tau y \supset y =_\tau z \supset a^\tau =_\tau z)}\ (\forall_I)$$
$$\overline{\forall x^\tau \forall y^\tau \forall z^\tau (x =_\tau y \supset y =_\tau z \supset x =_\tau z)}\ (\forall_I)$$

Derivation (a):

Left branch:

$$[b =_\tau c]^2$$
$$(Def.)\ \overline{(\lambda x^\tau \lambda y^\tau . \forall Y^{\tau\to o}(Y\ x \leftrightarrow Y\ y))\ b^\tau\ c^\tau}$$
$$(=_{\alpha\beta})\ \overline{\forall Y^{\tau\to o}(Y\ b^\tau \leftrightarrow Y\ c^\tau)}$$
$$(\forall_E)\ \overline{P^{\tau\to o}b^\tau \leftrightarrow P^{\tau\to o}c^\tau}$$
$$(Def.)\ \overline{(P^{\tau\to o}b^\tau \supset P^{\tau\to o}c^\tau) \wedge (P^{\tau\to o}c^\tau \supset P^{\tau\to o}b^\tau)}$$
$$(\wedge_E)\ \overline{P^{\tau\to o}b^\tau \supset P^{\tau\to o}c^\tau}$$

Right branch:

$$[a =_\tau b]^1$$
$$\overline{(\lambda x^\tau \lambda y^\tau . \forall Y^{\tau\to o}(Y\ x \leftrightarrow Y\ y))\ a^\tau\ b^\tau}\ (Def.)$$
$$\overline{\forall Y^{\tau\to o}(Y\ a^\tau \leftrightarrow Y\ b^\tau)}\ (=_{\alpha\beta})$$
$$\overline{P^{\tau\to o}a^\tau \leftrightarrow P^{\tau\to o}b^\tau}\ (\forall_E)$$
$$\overline{(P^{\tau\to o}a^\tau \supset P^{\tau\to o}b^\tau) \wedge (P^{\tau\to o}b^\tau \supset P^{\tau\to o}a^\tau)}\ (Def.)$$
$$\overline{P^{\tau\to o}a^\tau \supset P^{\tau\to o}b^\tau}\ (\wedge_E)$$
$$\frac{P^{\tau\to o}a^\tau \supset P^{\tau\to o}b^\tau \qquad [P^{\tau\to o}a^\tau]^{3a}}{P^{\tau\to o}b^\tau}\ (\supset_E)$$

Combined:

$$\frac{P^{\tau\to o}b^\tau \supset P^{\tau\to o}c^\tau \qquad P^{\tau\to o}b^\tau}{P^{\tau\to o}c^\tau}\ (\to_E)$$
$$\overline{(a)}\ (\supset_{3a})$$

Derivation (b):

Left branch:

$$[a =_\tau b]^1$$
$$(Def.)\ \overline{(\lambda x^\tau \lambda y^\tau . \forall Y^{\tau\to o}(Y\ x \leftrightarrow Y\ y))\ a^\tau\ b^\tau}$$
$$(=_{\alpha\beta})\ \overline{\forall Y^{\tau\to o}(Y\ a^\tau \leftrightarrow Y\ b^\tau)}$$
$$(\forall_E)\ \overline{P^{\tau\to o}a^\tau \leftrightarrow P^{\tau\to o}b^\tau}$$
$$(Def.)\ \overline{(P^{\tau\to o}a^\tau \supset P^{\tau\to o}b^\tau) \wedge (P^{\tau\to o}b^\tau \supset P^{\tau\to o}a^\tau)}$$
$$(\wedge_E)\ \overline{P^{\tau\to o}b^\tau \supset P^{\tau\to o}a^\tau}$$

Right branch:

$$[b =_\tau c]^2$$
$$\overline{(\lambda x^\tau \lambda y^\tau . \forall Y^{\tau\to o}(Y\ x \leftrightarrow Y\ y))\ b^\tau\ c^\tau}\ (Def.)$$
$$\overline{\forall Y^{\tau\to o}(Y\ b^\tau \leftrightarrow Y\ c^\tau)}\ (=_{\alpha\beta})$$
$$\overline{P^{\tau\to o}b^\tau \leftrightarrow P^{\tau\to o}c^\tau}\ (\forall_E)$$
$$\overline{(P^{\tau\to o}b^\tau \supset P^{\tau\to o}c^\tau) \wedge (P^{\tau\to o}c^\tau \supset P^{\tau\to o}b^\tau)}\ (Def.)$$
$$\overline{P^{\tau\to o}c^\tau \supset P^{\tau\to o}b^\tau}\ (\wedge_E)$$
$$\frac{P^{\tau\to o}c^\tau \supset P^{\tau\to o}b^\tau \qquad [P^{\tau\to o}c^\tau]^{3b}}{P^{\tau\to o}b^\tau}\ (\supset_E)$$

Combined:

$$\frac{P^{\tau\to o}b^\tau \supset P^{\tau\to o}a^\tau \qquad P^{\tau\to o}b^\tau}{P^{\tau\to o}a^\tau}\ (\to_E)$$
$$\overline{(b)}\ (\supset_{3b})$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[a =_\tau b]^1}{(\lambda x^\tau \lambda y^\tau.\forall Y^{\tau\to o}(Y\ x \leftrightarrow Y\ y))\ a^\tau\ b^\tau}\,(Def.)}}{\forall Y^{\tau\to o}(Y\ a^\tau \leftrightarrow Y\ b^\tau)}\,(=_{\alpha\beta})}}{(\lambda x^\tau.P^{\sigma\to o}(g^{\tau\to\sigma}x))\ a^\tau \leftrightarrow (\lambda x^\tau.P^{\sigma\to o}(g^{\tau\to\sigma}x))\ b^\tau}\,(\forall_E)}}{P^{\sigma\to o}(g^{\tau\to\sigma}a^\tau) \leftrightarrow P^{\sigma\to o}(g^{\tau\to\sigma}b^\tau)}\,(=_{\alpha\beta})}}{\forall Y^{\sigma\to o}(Y(g^{\tau\to\sigma}a^\tau) \leftrightarrow Y(g^{\tau\to\sigma}b^\tau))}\,(\forall_I)}}{(\lambda x^\sigma \lambda y^\sigma.\forall Y^{\sigma\to o}(Yx \leftrightarrow Yy))\ (g^{\tau\to\sigma}a^\tau)\ (g^{\tau\to\sigma}b^\tau)}\,(=_{\alpha\beta})}}{g^{\tau\to\sigma}a^\tau =_\sigma g^{\tau\to\sigma}b^\tau}\,(Def.)}}{a^\tau =_\tau b^\tau \supset g^{\tau\to\sigma}a^\tau =_\sigma g^{\tau\to\sigma}b^\tau}\,(\supset_I^1)}}{\forall y^\tau(a^\tau =_\tau y \supset g^{\tau\to\sigma}a^\tau =_\sigma g^{\tau\to\sigma}y)}\,(\forall_I)}}{\forall x^\tau \forall y^\tau(x =_\tau y \supset g^{\tau\to\sigma}x =_\sigma g^{\tau\to\sigma}y)}\,(\forall_I)}}{\forall f^{\tau\to\sigma}\forall x^\tau \forall y^\tau(x =_\tau y \supset fx =_\sigma fy)}\,(\forall_I)$$

Exercise ?.4

The set $\mathtt{fML}$ of formulae of (propositional) modal logic is defined as follows where $\mathcal{PV}$ is a countably infinite set of propositional variables.

$$\mathtt{fML} ::= \bot \mid p \in \mathcal{PV} \mid \mathtt{ML} \to \mathtt{ML} \mid \Box\mathtt{ML}$$

A Kripke frame is a pair (W, R) where W is a non-empty set and R is a binary relation on W. A Kripke model is a pair (F, V) where F is a Kripke frame and V is a valuation mapping each $p \in \mathcal{PV}$ to $V(p) \subseteq W$.

The satisfaction of a formula A on a Kripke model $((W, R), V)$ at $x \in W$ (denoted $((W, R), V), x \vDash A$) is defined inductively on the structure of A as follows. (NB. its negation is denoted $((W, R), V), x \nvDash A$).

- $(F, V), x \nvDash \bot$ never
- $(F, V), x \vDash p$ iff $x \in V(p)$
- $(F, V), x \vDash A \to B$ iff $(F, V), x \nvDash A$ or $(F, V), x \vDash B$
- $(F, V), x \vDash \Box A$ iff $(F, V), y \vDash A$ for every y such that Rxy

It is well-known that the set Γ of modal formulae (defined below) is exactly the set of theorems of the normal modal logic **K**.

$\{A \mid ((W, R), V), x \vDash A$ for every Kripke model $((W, R), V)$ and $x \in W\}$

Demonstrate that this is a second-order logic with respect to the semantics in Section 2 of the lecture notes. The modal logic **K** (and hence Λ) has a finite axiomatisation. How is this compatible with the "incompleteness of second-order logic" result that we established in class?

Consider the translation $\tau : \texttt{fML} \to \mathcal{L}$, where $\mathcal{L}$ is the set of second order formulas over the signature $\Sigma := \langle R/2 \rangle$, i.e. a signature with a single two-ary predicate, such that

$$\begin{aligned}
&\tau(\bot)[x] := \bot \\
&\tau(p)[x] := P(x) \text{ for } p \in \mathcal{PV} \\
&\tau(\neg\varphi)[x] := \neg\tau(\varphi)[x] \\
&\tau(\varphi \wedge \psi)[x] := \tau(\varphi)[x] \wedge \tau(\psi)[x] \\
&\tau(\varphi \vee \psi)[x] := \tau(\varphi)[x] \vee \tau(\psi)[x] \\
&\tau(\varphi \to \psi)[x] := \tau(\varphi)[x] \to \tau(\psi)[x] \\
&\tau(\Box\varphi)[x] := \forall y(R(x,y) \to \tau(\varphi))[y] \text{ with } y \text{ fresh}
\end{aligned}$$

where $[x]$ indicates that x is the only free variable in the formula. (In general let $[x_1, \ldots, x_m, P_1 \ldots P_n]$ indicate the free variables in a formula.)
Using τ one can define the mapping $\tau^* : \texttt{fML} \to \mathcal{L}$ which maps an arbitrary formula $\varphi \in \texttt{fML}$ to the formula $\tau^c(\varphi) \in \mathcal{L}$. With

$$\tau^c(\varphi) := \forall P_1 \ldots \forall P_n \forall x_1 \ldots \forall x_m \tau(\varphi[x_1, \ldots, x_m, P_1, \ldots, P_n])$$

That is, τ^c is simply the universal closure of τ, for the free variables $x_1, \ldots, x_m$ and the free second order monadic variables $P_1, \ldots, P_n$. First, notice that there can only be one free variable in φ. That is, in modal logic one starts to evaluate starting from a specific world and a world transition only happens via $\Box$, which given the translations only introduces bound variables. Hence,

$$\tau^c(\varphi) := \forall P_1 \ldots \forall P_n \forall x \, \tau(\varphi)[x, P_1, \ldots, P_n]$$

Alternatively, one can view s as free variable and only consider the following mapping

$$\tau^o(\varphi)[x] := \forall P_1 \ldots \forall P_n \, \tau(\varphi)[x, P_1, \ldots, P_n]$$

Note: For the sake of readability free predicate variables are suppressed if not required.
I choose to pursuit the proof on basis of τ^o, due to the fact that for example in **S5** there is a difference between local and global consequence, with local being stronger that global. (I took course in Dynamic Epistemic Logic) Hence, to avoid pitfalls I choose to focus on the local case and then argue for the global one. Even though, I would conjecture that in this case there is probably no difference.
In modal logic one often speaks of frames. That is, one abstracts away from the variable assignment of a model and speaks only of the underlying structure that is, for a model $\mathcal{M} := \langle W, R, V \rangle$ the corresponding frame is $\mathcal{M} := \langle W, R \rangle$. If a formula holds for every model over a specific W and a

corresponding R, then this formula will hold in a frame. Since **K** consists of formulas that hold in arbitrary models one can restate its definition as the set

$$\{A \mid \mathcal{F}, s \vDash A \text{ for every Kripke frame } \mathcal{F} \text{ and } s \in W\}$$

Furthermore, since it has to hold in every state it follows that this is the same as

$$\{A \mid \mathcal{F} \vDash A \text{ for every Kripke frame } \mathcal{F}\}$$

Moreover, we observe that for a frame $\mathcal{F} \coloneqq \langle W, R\rangle$ there exists a standard second order structure $\mathcal{I}_{\mathcal{F}} \coloneqq \langle D, I\rangle$, where I maps the symbol R to the relation of the frame, i.e. $I(R) \coloneqq R$. And vice versa. Hence, if not explicitly required I will not distinguish between those two and will write $\mathbb{F} \vDash_K \varphi$ for the inference in modal logic, and will write $\mathbb{F} \vDash \varphi$ for the inference $\mathcal{I}_{\mathcal{F}} \vDash \varphi$ in second order logic.
Similarly, for a model $\mathcal{M} \coloneqq \langle W, R, V\rangle$ of a frame $\mathcal{F} \coloneqq \langle W, R\rangle$. This frame has a corresponding interpretation $\mathcal{I}_{\mathcal{F}}$. Now using $V \coloneqq \{p_1 \mapsto S_1, \ldots, p_n \mapsto S_n, \}$ where $S_1, \ldots S_n \subseteq W$, one can construct an appropriate variable assignment $\{P_1 \mapsto V(p_1), \ldots, P_n \mapsto V(p_n)\}$ with witch to extend $\mathcal{I}_{\mathcal{F}}$ to construct $\mathcal{I}_{\mathcal{M}} \coloneqq \mathcal{I}_{\mathcal{F}} \cup \{P_1 \mapsto V(p_1), \ldots, P_n \mapsto V(p_n)\}$. Clearly, this works in the other direction as well. Moreover, as above if clear, the same name will reference both structure.
The aim is to show that for a formula $\varphi \in \texttt{fML}$ and for an arbitrary $\mathcal{F}$ and arbitrary $s \in W$

$$\mathcal{F}, s \vDash_K \varphi \iff \mathcal{F} \cup \{x \mapsto s\} \vDash \tau^c(\varphi)[x]$$

To do this it will first be shown by induction on a formula $\varphi \in \texttt{fML}$, that

$$\mathcal{M}, s \vDash_K \varphi \iff \mathcal{M} \cup \{x \mapsto s\} \vDash \tau(\varphi)[x]$$

IB: Let $\varphi \in \texttt{fML}$. There are two base cases to consider.

- if $\varphi = \bot$. $\mathcal{F}, s \vDash_K \bot$ can never be the case. Similarly, no structure in SOL models $\tau(\bot) = \bot$.
- if $\varphi = p$. Starting from $\mathcal{M}, s \vDash_K p$, which is equivalent to $s \in V(p)$. By construction $\tau(P)[x] = P(x)$ and $I(P) = V(p)$. Hence, $s \in I(P)$ and from $I(x) = s$ it follows that $I(x) \in I(P)$. Which by semantics is $\mathcal{M} \cup \{x \mapsto s\} \vDash P(x)$, which is the same as $\mathcal{M} \cup \{x \mapsto s\} \vDash \tau(p)[x]$.

IS: Let $\varphi \in \texttt{fML}$. There are two base cases to consider.

- $\varphi = \neg\psi$: Start from $\mathcal{M}, s \vDash_K \neg\psi$. By semantics $\mathcal{M}, s \nvDash_K \psi$. By IH $\mathcal{M} \cup \{x \mapsto s\} \nvDash \tau(\psi)[x]$. By semantics $\mathcal{M} \cup \{x \mapsto s\} \vDash \neg\tau(\psi)[x]$. By definition $\mathcal{M} \cup \{x \mapsto s\} \vDash \tau(\neg\psi)[x]$.

- $\varphi = \psi \wedge \chi$: Start from $\mathcal{M}, s \vDash_K \psi \wedge \chi$. By semantics $\mathcal{M}, s \vDash_K \psi$ *and* $\mathcal{M}, s \vDash_K \chi$. By IH $\mathcal{M} \cup \{x \mapsto s\} \vDash \tau(\psi)[x]$ *and* $\mathcal{M} \cup \{x \mapsto s\} \vDash \tau(\chi)[x]$. By semantics $\mathcal{M} \cup \{x \mapsto s\} \vDash \tau(\psi)[x] \wedge \tau(\chi)[x]$. By definition $\mathcal{M} \cup \{x \mapsto s\} \vDash \tau(\psi \wedge \chi)[x]$.

- $\varphi = \psi \vee \chi$: Start from $\mathcal{M}, s \vDash_K \psi \vee \chi$. By semantics $\mathcal{M}, s \vDash_K \psi$ *or* $\mathcal{M}, s \vDash_K \chi$. By IH $\mathcal{M} \cup \{x \mapsto s\} \vDash \tau(\psi)[x]$ *or* $\mathcal{M} \cup \{x \mapsto s\} \vDash \tau(\chi)[x]$. By semantics $\mathcal{M} \cup \{x \mapsto s\} \vDash \tau(\psi)[x] \vee \tau(\chi)[x]$. By definition $\mathcal{M} \cup \{x \mapsto s\} \vDash \tau(\psi \vee \chi)[x]$.

- $\varphi = \psi \to \chi$: Start from $\mathcal{M}, s \vDash_K \psi \to \chi$. By semantics $\mathcal{M}, s \vDash_K \psi \Rightarrow \mathcal{M}, s \vDash_K \chi$. By IH $\mathcal{M} \cup \{x \mapsto s\} \vDash \tau(\psi)[x] \Rightarrow \mathcal{M} \cup \{x \mapsto s\} \vDash \tau(\chi)[x]$. By semantics $\mathcal{M} \cup \{x \mapsto s\} \vDash \tau(\psi)[x] \to \tau(\chi)[x]$. By definition $\mathcal{M} \cup \{x \mapsto s\} \vDash \tau(\psi \to \chi)[x]$.

- $\varphi = \Box\psi$: Start from $\mathcal{M}, s \vDash_K \Box\psi$. By semantics $\forall t(R(s,t) \Rightarrow \mathcal{M}, t \vDash_K \psi)$. By IH $\forall t(R(s,t) \Rightarrow \mathcal{M} \cup \{y \mapsto t\} \vDash \tau(\psi)[y])$. By semantics (definition of $\mathcal{M} = \mathcal{I}_{\mathcal{M}}$), $\forall t(\mathcal{M} \cup \{x \mapsto s, y \mapsto t\} \vDash R(x,y) \to \tau(\psi)[y])$. By semantics $\mathcal{M} \cup \{x \mapsto s\} \vDash \forall y(R(x,y) \to \tau(\psi)[y])$. By definition $\mathcal{M} \cup \{x \mapsto s\} \vDash \tau(\Box\psi)[x]$.

Using this consider the statement $\mathcal{M} \vDash_K \varphi$ for any $\varphi \in \texttt{fML}$. This is the same as $\forall s \mathcal{M}, s \vDash_K \varphi$. Which by the above observation is equal to $\forall s \mathcal{M} \cup \{x \mapsto s\} \vDash \tau(\varphi)[x]$. Which incidentally is the semantics of $\mathcal{M} \vDash \forall s\, \tau(\varphi)$. Hence,

$$\mathcal{M} \vDash_K \varphi \iff \mathcal{M} \vDash \forall x\, \tau(\varphi)$$

Using this consider the statement $\mathcal{F}, s \vDash_K \varphi$ for any $\varphi \in \texttt{fML}$. This is the same as for all V, $\langle \mathcal{F}, V \rangle, s \vDash_K \varphi$. That is, $\mathcal{M}, s \vDash_K \varphi$ for some model of the frame $\mathcal{F}$. Now this is the same as $\mathcal{M} \cup \{x \mapsto s\} \vDash \tau(\varphi)[x]$, which is the same as for all V, $\mathcal{F} \cup \{x \mapsto s, P_1 \mapsto V(p_1), \ldots, P_n \mapsto V(p_n)\} \vDash \tau(\varphi)[x, P_1, \ldots, P_n]$. Which by semantics is $\mathcal{F} \cup \{x \mapsto s\} \vDash \forall P_1 \ldots \forall P_n\, \tau(\varphi)[x, P_1, \ldots, P_n]$, i.e. since variable assignment for p is simply a subset of W, thus one iterates over all of them to obtain all possible variable assignments. Hence,

$$\mathcal{F}, s \vDash_K \varphi \iff \mathcal{F} \cup \{x \mapsto s\} \vDash \forall P_1 \ldots \forall P_n\, \tau(\varphi)[x, P_1, \ldots, P_n] \iff \mathcal{F} \cup \{x \mapsto s\} \vDash \tau^o(\varphi)[x]$$

By similar reasoning as above and by quantifier shift

$$\mathcal{F} \vDash_K \varphi \iff \mathcal{F} \vDash \forall P_1 \ldots \forall P_n \forall x\, \tau(\varphi)[x, P_1, \ldots, P_n] \iff \mathcal{F} \vDash \tau^c(\varphi)$$

Hence, one obtains

$$\{\varphi \mid \mathcal{F} \vDash \varphi \text{ for every Kripke frame } \mathcal{F}\} = \{\varphi \mid \mathcal{F} \vDash \varphi \text{ for every structure } \mathcal{F}\}$$

> The modal logic **K** (and hence Λ) has a finite axiomatisation. How is this compatible with the "incompleteness of second-order logic" result that we established in class?

My conjecture would be, due to the fact that the language one operates over is just a fragment of full second order logic. That is, $\mathcal{L} \subset \mathcal{L}_{SO}$. Similar to the fact that $\mathcal{L}_{FO} \subset \mathcal{L}_{SO}$, meaning that one can use second order semantics to evaluate first order formulas, which are a fragment of the second order logic, and if one does so one regains completeness. Hence, in a similar vain if one investigates the transformation used, the only form of second order quantification is over sets. Hence, one deals with an MSO fragment of SOL. According to Daniel Leivant's Higher Order Logic from the course literature, MSO without function symbols is decidable. Similarly, in van Benthem the same (similar) is stated, that is that monadic second order predicate logic is decidable. This would explain that **K** is finitely axiomatiasible. Moreover, one would have a program that establishes if a formula is in **K**, taking this as a proof system would result in completeness. HOWEVER. Functions can be encoded as relations, and I am unsure if the presence of the relation symbol R results in a loss of this decidability.