Verbose proof for `strongPartRefReflexive`.

`strongPartRefReflexive:`

{1} FORALL pl, s: $(s \subseteq \{$——$\}(F(\mathrm{pl}))) \Rightarrow$ strongPartialRefinement(pl, pl, s)

`strongPartRefReflexive:`

{1} FORALL pl, s: $(s \subseteq \{$——$\}(F(\mathrm{pl}))) \Rightarrow$ strongPartialRefinement(pl, pl, s)

For the top quantifier in 1, we introduce Skolem constants: (pl s),

`strongPartRefReflexive:`

{1} $(s \subseteq \{$——$\}(F(\mathrm{pl}))) \Rightarrow$ strongPartialRefinement(pl, pl, s)

Expanding the definition of strongPartialRefinement,

`strongPartRefReflexive:`

{1} $(s \subseteq \{$——$\}(F(\mathrm{pl}))) \Rightarrow$
$(s \subseteq \{$——$\}(F(\mathrm{pl}))) \wedge$
$(s \subseteq \{$——$\}(F(\mathrm{pl}))) \wedge$ (FORALL c: $s(c) \Rightarrow$ (prod(pl, c) —- prod(pl, c)))

Applying bddsimp,

`strongPartRefReflexive:`

{-1} $(s \subseteq \{$——$\}(F(\mathrm{pl})))$

{1} FORALL c: $s(c) \Rightarrow$ (prod(pl, c) —- prod(pl, c))

Using lemma assetRefinement,

`strongPartRefReflexive:`

{-1} orders[set[Asset]].preorder?(—-)

{-2} $(s \subseteq \{$——$\}(F(\mathrm{pl})))$

{1} FORALL c: $s(c) \Rightarrow$ (prod(pl, c) —- prod(pl, c))

Expanding the definition of preorder?,

`strongPartRefReflexive:`

{-1} reflexive?(—-) & transitive?(—-)

{-2} $(s \subseteq \{$——$\}(F(\mathrm{pl})))$

{1} FORALL c: $s(c) \Rightarrow$ (prod(pl, c) —- prod(pl, c))

Applying disjunctive simplification to flatten sequent,

`strongPartRefReflexive:`

{-1}	reflexive?(—-)
{-2}	transitive?(—-)
{-3}	$(s \subseteq \{$——$\}(F(\mathrm{pl})))$
{1}	FORALL c: $s(c) \Rightarrow$ (prod(pl, c) —- prod(pl, c))

Expanding the definition of reflexive?,
`strongPartRefReflexive:`

{-1}	FORALL (x: set[Asset]): (x —- x)
{-2}	transitive?(—-)
{-3}	$(s \subseteq \{$——$\}(F(\mathrm{pl})))$
{1}	FORALL c: $s(c) \Rightarrow$ (prod(pl, c) —- prod(pl, c))

For the top quantifier in 1, we introduce Skolem constants: c,
`strongPartRefReflexive:`

{-1}	FORALL (x: set[Asset]): (x —- x)
{-2}	transitive?(—-)
{-3}	$(s \subseteq \{$——$\}(F(\mathrm{pl})))$
{1}	$s(c) \Rightarrow$ (prod(pl, c) —- prod(pl, c))

Instantiating the top quantifier in -1 with the terms: prod(pl, c),
`strongPartRefReflexive:`

{-1}	(prod(pl, c) —- prod(pl, c))
{-2}	transitive?(—-)
{-3}	$(s \subseteq \{$——$\}(F(\mathrm{pl})))$
{1}	$s(c) \Rightarrow$ (prod(pl, c) —- prod(pl, c))

Applying bddsimp,
This completes the proof of `strongPartRefReflexive`.
Q.E.D.

Verbose proof for `strongPartRefTransitive`.
`strongPartRefTransitive`:

$$\begin{array}{l|l}
& \\ \hline
\{1\} & (\textsc{forall}\ \mathrm{pl1},\ \mathrm{pl2},\ \mathrm{pl3},\ s,\ t: \\
& \quad (\mathrm{strongPartialRefinement}(\mathrm{pl1},\ \mathrm{pl2},\ s) \land \mathrm{strongPartialRefinement}(\mathrm{pl2},\ \mathrm{pl3},\ t)) \Rightarrow \\
& \quad\quad \mathrm{strongPartialRefinement}(\mathrm{pl1},\ \mathrm{pl3},\ (s \cap t)))
\end{array}$$

`strongPartRefTransitive`:

$$\begin{array}{l|l}
& \\ \hline
\{1\} & (\textsc{forall}\ \mathrm{pl1},\ \mathrm{pl2},\ \mathrm{pl3},\ s,\ t: \\
& \quad (\mathrm{strongPartialRefinement}(\mathrm{pl1},\ \mathrm{pl2},\ s) \land \mathrm{strongPartialRefinement}(\mathrm{pl2},\ \mathrm{pl3},\ t)) \Rightarrow \\
& \quad\quad \mathrm{strongPartialRefinement}(\mathrm{pl1},\ \mathrm{pl3},\ (s \cap t)))
\end{array}$$

For the top quantifier in 1, we introduce Skolem constants: (pl1 pl2 pl3 s t),
`strongPartRefTransitive`:

$$\begin{array}{l|l}
& \\ \hline
\{1\} & (\mathrm{strongPartialRefinement}(\mathrm{pl1},\ \mathrm{pl2},\ s) \land \mathrm{strongPartialRefinement}(\mathrm{pl2},\ \mathrm{pl3},\ t)) \Rightarrow \\
& \quad \mathrm{strongPartialRefinement}(\mathrm{pl1},\ \mathrm{pl3},\ (s \cap t))
\end{array}$$

Expanding the definition(s) of (strongPartialRefinement intersection),
`strongPartRefTransitive`:

$$\begin{array}{l|l}
& \\ \hline
\{1\} & (((s \subseteq \{\text{——}\}(F(\mathrm{pl1}))) \land \\
& \quad\quad (s \subseteq \{\text{——}\}(F(\mathrm{pl2}))) \land \\
& \quad\quad\quad (\textsc{forall}\ c:\ s(c) \Rightarrow (\mathrm{prod}(\mathrm{pl1},\ c)\ \text{—-}\ \mathrm{prod}(\mathrm{pl2},\ c)))) \\
& \quad \land \\
& \quad (t \subseteq \{\text{——}\}(F(\mathrm{pl2}))) \land \\
& \quad\quad (t \subseteq \{\text{——}\}(F(\mathrm{pl3}))) \land \\
& \quad\quad\quad (\textsc{forall}\ c:\ t(c) \Rightarrow (\mathrm{prod}(\mathrm{pl2},\ c)\ \text{—-}\ \mathrm{prod}(\mathrm{pl3},\ c)))) \\
& \Rightarrow \\
& (\{x \mid (x \in s) \land (x \in t)\} \subseteq \{\text{——}\}(F(\mathrm{pl1}))) \land \\
& \quad (\{x \mid (x \in s) \land (x \in t)\} \subseteq \{\text{——}\}(F(\mathrm{pl3}))) \land \\
& \quad\quad (\textsc{forall}\ c:\ (c \in s) \land (c \in t) \Rightarrow (\mathrm{prod}(\mathrm{pl1},\ c)\ \text{—-}\ \mathrm{prod}(\mathrm{pl3},\ c)))
\end{array}$$

Expanding the definition of member,

`strongPartRefTransitive:`

$$
\begin{array}{l}
\hline
\{1\}\quad (((s \subseteq \{\text{——}\}(F(\text{pl1}))) \;\wedge \\
\qquad\qquad (s \subseteq \{\text{——}\}(F(\text{pl2}))) \;\wedge \\
\qquad\qquad\; (\textsc{forall}\;\; c\text{:}\;\; s(c) \;\Rightarrow\; (\text{prod}(\text{pl1},\; c) \text{ —- } \text{prod}(\text{pl2},\; c)))) \\
\qquad\quad \wedge \\
\qquad\quad (t \subseteq \{\text{——}\}(F(\text{pl2}))) \;\wedge \\
\qquad\quad\; (t \subseteq \{\text{——}\}(F(\text{pl3}))) \;\wedge \\
\qquad\qquad (\textsc{forall}\;\; c\text{:}\;\; t(c) \;\Rightarrow\; (\text{prod}(\text{pl2},\; c) \text{ —- } \text{prod}(\text{pl3},\; c)))) \\
\qquad \Rightarrow \\
\qquad (\{x \mid s(x) \;\wedge\; t(x)\} \subseteq \{\text{——}\}(F(\text{pl1}))) \;\wedge \\
\qquad\; (\{x \mid s(x) \;\wedge\; t(x)\} \subseteq \{\text{——}\}(F(\text{pl3}))) \;\wedge \\
\qquad\quad (\textsc{forall}\;\; c\text{:}\;\; s(c) \;\wedge\; t(c) \;\Rightarrow\; (\text{prod}(\text{pl1},\; c) \text{ —- } \text{prod}(\text{pl3},\; c)))
\end{array}
$$

Expanding the definition of subset?,
`strongPartRefTransitive:`

$$
\begin{array}{l}
\hline
\{1\}\quad (((\textsc{forall}\;\; (x\text{:}\;\; \text{Conf})\text{:}\;\; (x \in s) \;\Rightarrow\; (x \in \{\text{——}\}(F(\text{pl1})))) \;\wedge \\
\qquad\qquad (\textsc{forall}\;\; (x\text{:}\;\; \text{Conf})\text{:}\;\; (x \in s) \;\Rightarrow\; (x \in \{\text{——}\}(F(\text{pl2})))) \;\wedge \\
\qquad\qquad\; (\textsc{forall}\;\; c\text{:}\;\; s(c) \;\Rightarrow\; (\text{prod}(\text{pl1},\; c) \text{ —- } \text{prod}(\text{pl2},\; c)))) \\
\qquad\quad \wedge \\
\qquad\quad (\textsc{forall}\;\; (x\text{:}\;\; \text{Conf})\text{:}\;\; (x \in t) \;\Rightarrow\; (x \in \{\text{——}\}(F(\text{pl2})))) \;\wedge \\
\qquad\quad\; (\textsc{forall}\;\; (x\text{:}\;\; \text{Conf})\text{:}\;\; (x \in t) \;\Rightarrow\; (x \in \{\text{——}\}(F(\text{pl3})))) \;\wedge \\
\qquad\qquad (\textsc{forall}\;\; c\text{:}\;\; t(c) \;\Rightarrow\; (\text{prod}(\text{pl2},\; c) \text{ —- } \text{prod}(\text{pl3},\; c)))) \\
\qquad \Rightarrow \\
\qquad (\textsc{forall}\;\; (\text{x\_1}\text{:}\;\; \text{Conf})\text{:} \\
\qquad\qquad (\text{x\_1} \in \{x \mid s(x) \;\wedge\; t(x)\}) \;\Rightarrow\; (\text{x\_1} \in \{\text{——}\}(F(\text{pl1})))) \\
\qquad\; \wedge \\
\qquad\; (\textsc{forall}\;\; (\text{x\_1}\text{:}\;\; \text{Conf})\text{:} \\
\qquad\qquad\; (\text{x\_1} \in \{x \mid s(x) \;\wedge\; t(x)\}) \;\Rightarrow\; (\text{x\_1} \in \{\text{——}\}(F(\text{pl3})))) \\
\qquad\quad \wedge\; (\textsc{forall}\;\; c\text{:}\;\; s(c) \;\wedge\; t(c) \;\Rightarrow\; (\text{prod}(\text{pl1},\; c) \text{ —- } \text{prod}(\text{pl3},\; c)))
\end{array}
$$

Expanding the definition of member,

strongPartRefTransitive:

$$
\begin{array}{l|l}
 & \\
\hline
\{1\} & (((\textsc{forall}\ (x\colon \text{Conf})\colon\ s(x)\ \Rightarrow\ \{\text{——}\}(F(\text{pl1}))(x))\ \wedge \\
 & \qquad (\textsc{forall}\ (x\colon \text{Conf})\colon\ s(x)\ \Rightarrow\ \{\text{——}\}(F(\text{pl2}))(x))\ \wedge \\
 & \qquad\ (\textsc{forall}\ c\colon\ s(c)\ \Rightarrow\ (\text{prod}(\text{pl1},\ c)\ \text{—-}\ \text{prod}(\text{pl2},\ c)))) \\
 & \quad \wedge \\
 & \quad (\textsc{forall}\ (x\colon \text{Conf})\colon\ t(x)\ \Rightarrow\ \{\text{——}\}(F(\text{pl2}))(x))\ \wedge \\
 & \quad\ (\textsc{forall}\ (x\colon \text{Conf})\colon\ t(x)\ \Rightarrow\ \{\text{——}\}(F(\text{pl3}))(x))\ \wedge \\
 & \qquad (\textsc{forall}\ c\colon\ t(c)\ \Rightarrow\ (\text{prod}(\text{pl2},\ c)\ \text{—-}\ \text{prod}(\text{pl3},\ c)))) \\
 & \Rightarrow \\
 & (\textsc{forall}\ (\text{x\_1}\colon \text{Conf})\colon\ s(\text{x\_1})\ \wedge\ t(\text{x\_1})\ \Rightarrow\ \{\text{——}\}(F(\text{pl1}))(\text{x\_1}))\ \wedge \\
 & \ (\textsc{forall}\ (\text{x\_1}\colon \text{Conf})\colon\ s(\text{x\_1})\ \wedge\ t(\text{x\_1})\ \Rightarrow\ \{\text{——}\}(F(\text{pl3}))(\text{x\_1}))\ \wedge \\
 & \quad (\textsc{forall}\ c\colon\ s(c)\ \wedge\ t(c)\ \Rightarrow\ (\text{prod}(\text{pl1},\ c)\ \text{—-}\ \text{prod}(\text{pl3},\ c)))
\end{array}
$$

Applying bddsimp,
we get 3 subgoals:
strongPartRefTransitive.1:

$$
\begin{array}{l|l}
\{-1\} & \textsc{forall}\ (x\colon \text{Conf})\colon\ s(x)\ \Rightarrow\ \{\text{——}\}(F(\text{pl1}))(x) \\
\{-2\} & \textsc{forall}\ (x\colon \text{Conf})\colon\ s(x)\ \Rightarrow\ \{\text{——}\}(F(\text{pl2}))(x) \\
\{-3\} & \textsc{forall}\ c\colon\ s(c)\ \Rightarrow\ (\text{prod}(\text{pl1},\ c)\ \text{—-}\ \text{prod}(\text{pl2},\ c)) \\
\{-4\} & \textsc{forall}\ (x\colon \text{Conf})\colon\ t(x)\ \Rightarrow\ \{\text{——}\}(F(\text{pl2}))(x) \\
\{-5\} & \textsc{forall}\ (x\colon \text{Conf})\colon\ t(x)\ \Rightarrow\ \{\text{——}\}(F(\text{pl3}))(x) \\
\{-6\} & \textsc{forall}\ c\colon\ t(c)\ \Rightarrow\ (\text{prod}(\text{pl2},\ c)\ \text{—-}\ \text{prod}(\text{pl3},\ c)) \\
\hline
\{1\} & \textsc{forall}\ c\colon\ s(c)\ \wedge\ t(c)\ \Rightarrow\ (\text{prod}(\text{pl1},\ c)\ \text{—-}\ \text{prod}(\text{pl3},\ c))
\end{array}
$$

For the top quantifier in 1, we introduce Skolem constants: c,
strongPartRefTransitive.1:

$$
\begin{array}{l|l}
\{-1\} & \textsc{forall}\ (x\colon \text{Conf})\colon\ s(x)\ \Rightarrow\ \{\text{——}\}(F(\text{pl1}))(x) \\
\{-2\} & \textsc{forall}\ (x\colon \text{Conf})\colon\ s(x)\ \Rightarrow\ \{\text{——}\}(F(\text{pl2}))(x) \\
\{-3\} & \textsc{forall}\ c\colon\ s(c)\ \Rightarrow\ (\text{prod}(\text{pl1},\ c)\ \text{—-}\ \text{prod}(\text{pl2},\ c)) \\
\{-4\} & \textsc{forall}\ (x\colon \text{Conf})\colon\ t(x)\ \Rightarrow\ \{\text{——}\}(F(\text{pl2}))(x) \\
\{-5\} & \textsc{forall}\ (x\colon \text{Conf})\colon\ t(x)\ \Rightarrow\ \{\text{——}\}(F(\text{pl3}))(x) \\
\{-6\} & \textsc{forall}\ c\colon\ t(c)\ \Rightarrow\ (\text{prod}(\text{pl2},\ c)\ \text{—-}\ \text{prod}(\text{pl3},\ c)) \\
\hline
\{1\} & s(c)\ \wedge\ t(c)\ \Rightarrow\ (\text{prod}(\text{pl1},\ c)\ \text{—-}\ \text{prod}(\text{pl3},\ c))
\end{array}
$$

Instantiating the top quantifier in -1 with the terms: c,

`strongPartRefTransitive.1:`

{-1}	$s(c) \Rightarrow \{$——$\}(F(\text{pl1}))(c)$
{-2}	FORALL $(x\colon$ Conf$)\colon s(x) \Rightarrow \{$——$\}(F(\text{pl2}))(x)$
{-3}	FORALL $c\colon s(c) \Rightarrow (\text{prod}(\text{pl1},\ c)$ —- $\text{prod}(\text{pl2},\ c))$
{-4}	FORALL $(x\colon$ Conf$)\colon t(x) \Rightarrow \{$——$\}(F(\text{pl2}))(x)$
{-5}	FORALL $(x\colon$ Conf$)\colon t(x) \Rightarrow \{$——$\}(F(\text{pl3}))(x)$
{-6}	FORALL $c\colon t(c) \Rightarrow (\text{prod}(\text{pl2},\ c)$ —- $\text{prod}(\text{pl3},\ c))$
{1}	$s(c) \wedge t(c) \Rightarrow (\text{prod}(\text{pl1},\ c)$ —- $\text{prod}(\text{pl3},\ c))$

Instantiating the top quantifier in -2 with the terms: c,
`strongPartRefTransitive.1:`

{-1}	$s(c) \Rightarrow \{$——$\}(F(\text{pl1}))(c)$
{-2}	$s(c) \Rightarrow \{$——$\}(F(\text{pl2}))(c)$
{-3}	FORALL $c\colon s(c) \Rightarrow (\text{prod}(\text{pl1},\ c)$ —- $\text{prod}(\text{pl2},\ c))$
{-4}	FORALL $(x\colon$ Conf$)\colon t(x) \Rightarrow \{$——$\}(F(\text{pl2}))(x)$
{-5}	FORALL $(x\colon$ Conf$)\colon t(x) \Rightarrow \{$——$\}(F(\text{pl3}))(x)$
{-6}	FORALL $c\colon t(c) \Rightarrow (\text{prod}(\text{pl2},\ c)$ —- $\text{prod}(\text{pl3},\ c))$
{1}	$s(c) \wedge t(c) \Rightarrow (\text{prod}(\text{pl1},\ c)$ —- $\text{prod}(\text{pl3},\ c))$

Using lemma assetRefinement,
`strongPartRefTransitive.1:`

{-1}	orders[set[Asset]].preorder?(—-)
{-2}	$s(c) \Rightarrow \{$——$\}(F(\text{pl1}))(c)$
{-3}	$s(c) \Rightarrow \{$——$\}(F(\text{pl2}))(c)$
{-4}	FORALL $c\colon s(c) \Rightarrow (\text{prod}(\text{pl1},\ c)$ —- $\text{prod}(\text{pl2},\ c))$
{-5}	FORALL $(x\colon$ Conf$)\colon t(x) \Rightarrow \{$——$\}(F(\text{pl2}))(x)$
{-6}	FORALL $(x\colon$ Conf$)\colon t(x) \Rightarrow \{$——$\}(F(\text{pl3}))(x)$
{-7}	FORALL $c\colon t(c) \Rightarrow (\text{prod}(\text{pl2},\ c)$ —- $\text{prod}(\text{pl3},\ c))$
{1}	$s(c) \wedge t(c) \Rightarrow (\text{prod}(\text{pl1},\ c)$ —- $\text{prod}(\text{pl3},\ c))$

Expanding the definition of preorder?,
`strongPartRefTransitive.1:`

{-1}	reflexive?(—-) & transitive?(—-)
{-2}	$s(c) \Rightarrow \{$——$\}(F(\text{pl1}))(c)$
{-3}	$s(c) \Rightarrow \{$——$\}(F(\text{pl2}))(c)$
{-4}	FORALL $c\colon s(c) \Rightarrow (\text{prod}(\text{pl1},\ c)$ —- $\text{prod}(\text{pl2},\ c))$
{-5}	FORALL $(x\colon$ Conf$)\colon t(x) \Rightarrow \{$——$\}(F(\text{pl2}))(x)$
{-6}	FORALL $(x\colon$ Conf$)\colon t(x) \Rightarrow \{$——$\}(F(\text{pl3}))(x)$
{-7}	FORALL $c\colon t(c) \Rightarrow (\text{prod}(\text{pl2},\ c)$ —- $\text{prod}(\text{pl3},\ c))$
{1}	$s(c) \wedge t(c) \Rightarrow (\text{prod}(\text{pl1},\ c)$ —- $\text{prod}(\text{pl3},\ c))$

Applying disjunctive simplification to flatten sequent,

`strongPartRefTransitive.1:`

{-1}	reflexive?(—-)
{-2}	transitive?(—-)
{-3}	$s(c) \Rightarrow$ {——}(F(pl1))(c)
{-4}	$s(c) \Rightarrow$ {——}(F(pl2))(c)
{-5}	FORALL c: $s(c) \Rightarrow$ (prod(pl1, c) —- prod(pl2, c))
{-6}	FORALL (x: Conf): $t(x) \Rightarrow$ {——}(F(pl2))(x)
{-7}	FORALL (x: Conf): $t(x) \Rightarrow$ {——}(F(pl3))(x)
{-8}	FORALL c: $t(c) \Rightarrow$ (prod(pl2, c) —- prod(pl3, c))
{-9}	$s(c)$
{-10}	$t(c)$
{1}	(prod(pl1, c) —- prod(pl3, c))

Expanding the definition of transitive?,
`strongPartRefTransitive.1:`

{-1}	reflexive?(—-)
{-2}	FORALL (x: set[Asset]), (y: set[Asset]), (z: set[Asset]): (x —- y) & (y —- z) $\Rightarrow$ (x —- z)
{-3}	$s(c) \Rightarrow$ {——}(F(pl1))(c)
{-4}	$s(c) \Rightarrow$ {——}(F(pl2))(c)
{-5}	FORALL c: $s(c) \Rightarrow$ (prod(pl1, c) —- prod(pl2, c))
{-6}	FORALL (x: Conf): $t(x) \Rightarrow$ {——}(F(pl2))(x)
{-7}	FORALL (x: Conf): $t(x) \Rightarrow$ {——}(F(pl3))(x)
{-8}	FORALL c: $t(c) \Rightarrow$ (prod(pl2, c) —- prod(pl3, c))
{-9}	$s(c)$
{-10}	$t(c)$
{1}	(prod(pl1, c) —- prod(pl3, c))

Instantiating the top quantifier in -2 with the terms: prod(pl1, c), prod(pl2, c), prod(pl3, c),

`strongPartRefTransitive.1:`

{-1}	reflexive?(—-)
{-2}	(prod(pl1, c) —- prod(pl2, c)) & (prod(pl2, c) —- prod(pl3, c)) $\Rightarrow$ (prod(pl1, c) —- prod(pl3, c))
{-3}	$s(c) \Rightarrow$ {——}(F(pl1))(c)
{-4}	$s(c) \Rightarrow$ {——}(F(pl2))(c)
{-5}	FORALL c: $s(c) \Rightarrow$ (prod(pl1, c) —- prod(pl2, c))
{-6}	FORALL (x: Conf): $t(x) \Rightarrow$ {——}(F(pl2))(x)
{-7}	FORALL (x: Conf): $t(x) \Rightarrow$ {——}(F(pl3))(x)
{-8}	FORALL c: $t(c) \Rightarrow$ (prod(pl2, c) —- prod(pl3, c))
{-9}	$s(c)$
{-10}	$t(c)$
{1}	(prod(pl1, c) —- prod(pl3, c))

Applying bddsimp,
we get 2 subgoals:
`strongPartRefTransitive.1.1:`

{-1}	reflexive?(—-)
{-2}	$s(c)$
{-3}	{——}(F(pl1))(c)
{-4}	{——}(F(pl2))(c)
{-5}	FORALL c: $s(c) \Rightarrow$ (prod(pl1, c) —- prod(pl2, c))
{-6}	FORALL (x: Conf): $t(x) \Rightarrow$ {——}(F(pl2))(x)
{-7}	FORALL (x: Conf): $t(x) \Rightarrow$ {——}(F(pl3))(x)
{-8}	FORALL c: $t(c) \Rightarrow$ (prod(pl2, c) —- prod(pl3, c))
{-9}	$t(c)$
{1}	(prod(pl2, c) —- prod(pl3, c))
{2}	(prod(pl1, c) —- prod(pl3, c))

Instantiating the top quantifier in -5 with the terms: c,
`strongPartRefTransitive.1.1:`

{-1}	reflexive?(—-)
{-2}	$s(c)$
{-3}	{——}(F(pl1))(c)
{-4}	{——}(F(pl2))(c)
{-5}	$s(c) \Rightarrow$ (prod(pl1, c) —- prod(pl2, c))
{-6}	FORALL (x: Conf): $t(x) \Rightarrow$ {——}(F(pl2))(x)
{-7}	FORALL (x: Conf): $t(x) \Rightarrow$ {——}(F(pl3))(x)
{-8}	FORALL c: $t(c) \Rightarrow$ (prod(pl2, c) —- prod(pl3, c))
{-9}	$t(c)$
{1}	(prod(pl2, c) —- prod(pl3, c))
{2}	(prod(pl1, c) —- prod(pl3, c))

Instantiating the top quantifier in -8 with the terms: c,
`strongPartRefTransitive.1.1:`

{-1}	reflexive?(—-)
{-2}	$s(c)$
{-3}	{——}(F(pl1))(c)
{-4}	{——}(F(pl2))(c)
{-5}	$s(c)$ ⇒ (prod(pl1, c) —- prod(pl2, c))
{-6}	FORALL (x: Conf): $t(x)$ ⇒ {——}(F(pl2))(x)
{-7}	FORALL (x: Conf): $t(x)$ ⇒ {——}(F(pl3))(x)
{-8}	$t(c)$ ⇒ (prod(pl2, c) —- prod(pl3, c))
{-9}	$t(c)$
{1}	(prod(pl2, c) —- prod(pl3, c))
{2}	(prod(pl1, c) —- prod(pl3, c))

Applying bddsimp,
This completes the proof of `strongPartRefTransitive.1.1`.
`strongPartRefTransitive.1.2:`

{-1}	reflexive?(—-)
{-2}	$s(c)$
{-3}	{——}(F(pl1))(c)
{-4}	{——}(F(pl2))(c)
{-5}	FORALL c: $s(c)$ ⇒ (prod(pl1, c) —- prod(pl2, c))
{-6}	FORALL (x: Conf): $t(x)$ ⇒ {——}(F(pl2))(x)
{-7}	FORALL (x: Conf): $t(x)$ ⇒ {——}(F(pl3))(x)
{-8}	FORALL c: $t(c)$ ⇒ (prod(pl2, c) —- prod(pl3, c))
{-9}	$t(c)$
{1}	(prod(pl1, c) —- prod(pl2, c))
{2}	(prod(pl1, c) —- prod(pl3, c))

Using lemma assetRefinement,

`strongPartRefTransitive.1.2:`

{-1}	orders[set[Asset]].preorder?(—-)
{-2}	reflexive?(—-)
{-3}	$s(c)$
{-4}	{——}(F(pl1))(c)
{-5}	{——}(F(pl2))(c)
{-6}	FORALL c: $s(c)$ ⇒ (prod(pl1, c) —- prod(pl2, c))
{-7}	FORALL (x: Conf): $t(x)$ ⇒ {——}(F(pl2))(x)
{-8}	FORALL (x: Conf): $t(x)$ ⇒ {——}(F(pl3))(x)
{-9}	FORALL c: $t(c)$ ⇒ (prod(pl2, c) —- prod(pl3, c))
{-10}	$t(c)$
{1}	(prod(pl1, c) —- prod(pl2, c))
{2}	(prod(pl1, c) —- prod(pl3, c))

Expanding the definition of preorder?,
`strongPartRefTransitive.1.2:`

{-1}	reflexive?(—-) & transitive?(—-)
{-2}	reflexive?(—-)
{-3}	$s(c)$
{-4}	{——}(F(pl1))(c)
{-5}	{——}(F(pl2))(c)
{-6}	FORALL c: $s(c)$ ⇒ (prod(pl1, c) —- prod(pl2, c))
{-7}	FORALL (x: Conf): $t(x)$ ⇒ {——}(F(pl2))(x)
{-8}	FORALL (x: Conf): $t(x)$ ⇒ {——}(F(pl3))(x)
{-9}	FORALL c: $t(c)$ ⇒ (prod(pl2, c) —- prod(pl3, c))
{-10}	$t(c)$
{1}	(prod(pl1, c) —- prod(pl2, c))
{2}	(prod(pl1, c) —- prod(pl3, c))

Applying disjunctive simplification to flatten sequent,

`strongPartRefTransitive.1.2:`

{-1}	reflexive?(—-)
{-2}	transitive?(—-)
{-3}	reflexive?(—-)
{-4}	$s(c)$
{-5}	{——}(F(pl1))(c)
{-6}	{——}(F(pl2))(c)
{-7}	FORALL c: $s(c)$ ⇒ (prod(pl1, c) —- prod(pl2, c))
{-8}	FORALL (x: Conf): $t(x)$ ⇒ {——}(F(pl2))(x)
{-9}	FORALL (x: Conf): $t(x)$ ⇒ {——}(F(pl3))(x)
{-10}	FORALL c: $t(c)$ ⇒ (prod(pl2, c) —- prod(pl3, c))
{-11}	$t(c)$
{1}	(prod(pl1, c) —- prod(pl2, c))
{2}	(prod(pl1, c) —- prod(pl3, c))

Expanding the definition of transitive?,
`strongPartRefTransitive.1.2:`

{-1}	reflexive?(—-)
{-2}	FORALL (x: set[Asset]), (y: set[Asset]), (z: set[Asset]): (x —- y) & (y —- z) ⇒ (x —- z)
{-3}	reflexive?(—-)
{-4}	$s(c)$
{-5}	{——}(F(pl1))(c)
{-6}	{——}(F(pl2))(c)
{-7}	FORALL c: $s(c)$ ⇒ (prod(pl1, c) —- prod(pl2, c))
{-8}	FORALL (x: Conf): $t(x)$ ⇒ {——}(F(pl2))(x)
{-9}	FORALL (x: Conf): $t(x)$ ⇒ {——}(F(pl3))(x)
{-10}	FORALL c: $t(c)$ ⇒ (prod(pl2, c) —- prod(pl3, c))
{-11}	$t(c)$
{1}	(prod(pl1, c) —- prod(pl2, c))
{2}	(prod(pl1, c) —- prod(pl3, c))

Instantiating the top quantifier in -2 with the terms: prod(pl1, c), prod(pl2, c), prod(pl3, c),

`strongPartRefTransitive.1.2:`

{-1}	reflexive?(—-)
{-2}	(prod(pl1, c) —- prod(pl2, c)) & (prod(pl2, c) —- prod(pl3, c)) $\Rightarrow$ (prod(pl1, c) —- prod(pl3, c))
{-3}	reflexive?(—-)
{-4}	$s(c)$
{-5}	{——}(F(pl1))(c)
{-6}	{——}(F(pl2))(c)
{-7}	FORALL c: $s(c)$ $\Rightarrow$ (prod(pl1, c) —- prod(pl2, c))
{-8}	FORALL (x: Conf): $t(x)$ $\Rightarrow$ {——}(F(pl2))(x)
{-9}	FORALL (x: Conf): $t(x)$ $\Rightarrow$ {——}(F(pl3))(x)
{-10}	FORALL c: $t(c)$ $\Rightarrow$ (prod(pl2, c) —- prod(pl3, c))
{-11}	$t(c)$
{1}	(prod(pl1, c) —- prod(pl2, c))
{2}	(prod(pl1, c) —- prod(pl3, c))

Applying bddsimp,
`strongPartRefTransitive.1.2:`

{-1}	reflexive?(—-)
{-2}	$s(c)$
{-3}	{——}(F(pl1))(c)
{-4}	{——}(F(pl2))(c)
{-5}	FORALL c: $s(c)$ $\Rightarrow$ (prod(pl1, c) —- prod(pl2, c))
{-6}	FORALL (x: Conf): $t(x)$ $\Rightarrow$ {——}(F(pl2))(x)
{-7}	FORALL (x: Conf): $t(x)$ $\Rightarrow$ {——}(F(pl3))(x)
{-8}	FORALL c: $t(c)$ $\Rightarrow$ (prod(pl2, c) —- prod(pl3, c))
{-9}	$t(c)$
{1}	(prod(pl1, c) —- prod(pl2, c))
{2}	(prod(pl1, c) —- prod(pl3, c))

Instantiating the top quantifier in -5 with the terms: c,

`strongPartRefTransitive.1.2:`

{-1}	reflexive?(—-)
{-2}	$s(c)$
{-3}	{——}(F(pl1))(c)
{-4}	{——}(F(pl2))(c)
{-5}	$s(c)$ ⇒ (prod(pl1, c) —- prod(pl2, c))
{-6}	FORALL (x: Conf): $t(x)$ ⇒ {——}(F(pl2))(x)
{-7}	FORALL (x: Conf): $t(x)$ ⇒ {——}(F(pl3))(x)
{-8}	FORALL c: $t(c)$ ⇒ (prod(pl2, c) —- prod(pl3, c))
{-9}	$t(c)$
{1}	(prod(pl1, c) —- prod(pl2, c))
{2}	(prod(pl1, c) —- prod(pl3, c))

Applying bddsimp,
This completes the proof of `strongPartRefTransitive.1.2`.
`strongPartRefTransitive.2:`

{-1}	FORALL (x: Conf): $s(x)$ ⇒ {——}(F(pl1))(x)
{-2}	FORALL (x: Conf): $s(x)$ ⇒ {——}(F(pl2))(x)
{-3}	FORALL c: $s(c)$ ⇒ (prod(pl1, c) —- prod(pl2, c))
{-4}	FORALL (x: Conf): $t(x)$ ⇒ {——}(F(pl2))(x)
{-5}	FORALL (x: Conf): $t(x)$ ⇒ {——}(F(pl3))(x)
{-6}	FORALL c: $t(c)$ ⇒ (prod(pl2, c) —- prod(pl3, c))
{1}	FORALL (x_1: Conf): s(x_1) ∧ t(x_1) ⇒ {——}(F(pl3))(x_1)

For the top quantifier in 1, we introduce Skolem constants: c,
`strongPartRefTransitive.2:`

{-1}	FORALL (x: Conf): $s(x)$ ⇒ {——}(F(pl1))(x)
{-2}	FORALL (x: Conf): $s(x)$ ⇒ {——}(F(pl2))(x)
{-3}	FORALL c: $s(c)$ ⇒ (prod(pl1, c) —- prod(pl2, c))
{-4}	FORALL (x: Conf): $t(x)$ ⇒ {——}(F(pl2))(x)
{-5}	FORALL (x: Conf): $t(x)$ ⇒ {——}(F(pl3))(x)
{-6}	FORALL c: $t(c)$ ⇒ (prod(pl2, c) —- prod(pl3, c))
{1}	$s(c)$ ∧ $t(c)$ ⇒ {——}(F(pl3))(c)

Instantiating the top quantifier in -2 with the terms: c,

`strongPartRefTransitive.2`:

{-1}	FORALL (x: Conf): $s(x)$ ⇒ {——}(F(pl1))(x)
{-2}	$s(c)$ ⇒ {——}(F(pl2))(c)
{-3}	FORALL c: $s(c)$ ⇒ (prod(pl1, c) —- prod(pl2, c))
{-4}	FORALL (x: Conf): $t(x)$ ⇒ {——}(F(pl2))(x)
{-5}	FORALL (x: Conf): $t(x)$ ⇒ {——}(F(pl3))(x)
{-6}	FORALL c: $t(c)$ ⇒ (prod(pl2, c) —- prod(pl3, c))
{1}	$s(c) \wedge t(c)$ ⇒ {——}(F(pl3))(c)

Applying bddsimp,
`strongPartRefTransitive.2`:

{-1}	FORALL (x: Conf): $s(x)$ ⇒ {——}(F(pl1))(x)
{-2}	$s(c)$
{-3}	{——}(F(pl2))(c)
{-4}	FORALL c: $s(c)$ ⇒ (prod(pl1, c) —- prod(pl2, c))
{-5}	FORALL (x: Conf): $t(x)$ ⇒ {——}(F(pl2))(x)
{-6}	FORALL (x: Conf): $t(x)$ ⇒ {——}(F(pl3))(x)
{-7}	FORALL c: $t(c)$ ⇒ (prod(pl2, c) —- prod(pl3, c))
{-8}	$t(c)$
{1}	{——}(F(pl3))(c)

Instantiating the top quantifier in -6 with the terms: c,
`strongPartRefTransitive.2`:

{-1}	FORALL (x: Conf): $s(x)$ ⇒ {——}(F(pl1))(x)
{-2}	$s(c)$
{-3}	{——}(F(pl2))(c)
{-4}	FORALL c: $s(c)$ ⇒ (prod(pl1, c) —- prod(pl2, c))
{-5}	FORALL (x: Conf): $t(x)$ ⇒ {——}(F(pl2))(x)
{-6}	$t(c)$ ⇒ {——}(F(pl3))(c)
{-7}	FORALL c: $t(c)$ ⇒ (prod(pl2, c) —- prod(pl3, c))
{-8}	$t(c)$
{1}	{——}(F(pl3))(c)

Applying bddsimp,
This completes the proof of `strongPartRefTransitive.2`.

`strongPartRefTransitive.3`:

$$
\begin{array}{|ll}
\{-1\} & \text{FORALL } (x\text{: Conf})\text{: } s(x) \Rightarrow \{\text{——}\}(F(\text{pl1}))(x) \\
\{-2\} & \text{FORALL } (x\text{: Conf})\text{: } s(x) \Rightarrow \{\text{——}\}(F(\text{pl2}))(x) \\
\{-3\} & \text{FORALL } c\text{: } s(c) \Rightarrow (\text{prod}(\text{pl1}, c) \text{—-} \text{prod}(\text{pl2}, c)) \\
\{-4\} & \text{FORALL } (x\text{: Conf})\text{: } t(x) \Rightarrow \{\text{——}\}(F(\text{pl2}))(x) \\
\{-5\} & \text{FORALL } (x\text{: Conf})\text{: } t(x) \Rightarrow \{\text{——}\}(F(\text{pl3}))(x) \\
\{-6\} & \text{FORALL } c\text{: } t(c) \Rightarrow (\text{prod}(\text{pl2}, c) \text{—-} \text{prod}(\text{pl3}, c)) \\
\hline
\{1\} & \text{FORALL } (\text{x\_1: Conf})\text{: } s(\text{x\_1}) \wedge t(\text{x\_1}) \Rightarrow \{\text{——}\}(F(\text{pl1}))(\text{x\_1})
\end{array}
$$

For the top quantifier in 1, we introduce Skolem constants: c,
`strongPartRefTransitive.3`:

$$
\begin{array}{|ll}
\{-1\} & \text{FORALL } (x\text{: Conf})\text{: } s(x) \Rightarrow \{\text{——}\}(F(\text{pl1}))(x) \\
\{-2\} & \text{FORALL } (x\text{: Conf})\text{: } s(x) \Rightarrow \{\text{——}\}(F(\text{pl2}))(x) \\
\{-3\} & \text{FORALL } c\text{: } s(c) \Rightarrow (\text{prod}(\text{pl1}, c) \text{—-} \text{prod}(\text{pl2}, c)) \\
\{-4\} & \text{FORALL } (x\text{: Conf})\text{: } t(x) \Rightarrow \{\text{——}\}(F(\text{pl2}))(x) \\
\{-5\} & \text{FORALL } (x\text{: Conf})\text{: } t(x) \Rightarrow \{\text{——}\}(F(\text{pl3}))(x) \\
\{-6\} & \text{FORALL } c\text{: } t(c) \Rightarrow (\text{prod}(\text{pl2}, c) \text{—-} \text{prod}(\text{pl3}, c)) \\
\hline
\{1\} & s(c) \wedge t(c) \Rightarrow \{\text{——}\}(F(\text{pl1}))(c)
\end{array}
$$

Instantiating the top quantifier in -1 with the terms: c,
`strongPartRefTransitive.3`:

$$
\begin{array}{|ll}
\{-1\} & s(c) \Rightarrow \{\text{——}\}(F(\text{pl1}))(c) \\
\{-2\} & \text{FORALL } (x\text{: Conf})\text{: } s(x) \Rightarrow \{\text{——}\}(F(\text{pl2}))(x) \\
\{-3\} & \text{FORALL } c\text{: } s(c) \Rightarrow (\text{prod}(\text{pl1}, c) \text{—-} \text{prod}(\text{pl2}, c)) \\
\{-4\} & \text{FORALL } (x\text{: Conf})\text{: } t(x) \Rightarrow \{\text{——}\}(F(\text{pl2}))(x) \\
\{-5\} & \text{FORALL } (x\text{: Conf})\text{: } t(x) \Rightarrow \{\text{——}\}(F(\text{pl3}))(x) \\
\{-6\} & \text{FORALL } c\text{: } t(c) \Rightarrow (\text{prod}(\text{pl2}, c) \text{—-} \text{prod}(\text{pl3}, c)) \\
\hline
\{1\} & s(c) \wedge t(c) \Rightarrow \{\text{——}\}(F(\text{pl1}))(c)
\end{array}
$$

Applying bddsimp,
This completes the proof of `strongPartRefTransitive.3`.
Q.E.D.

Verbose proof for `fmCompStrongDef`.

`fmCompStrongDef`:

$$\begin{array}{l} \hline \{1\} \quad \text{FORALL } (\mathrm{pl},\ \mathrm{fm2},\ s): \\ \quad (\mathrm{fmPartialRefinement}(F(\mathrm{pl}),\ \mathrm{fm2},\ s) \wedge \mathrm{wfPL}(\mathrm{pl2}) \Rightarrow \\ \quad\quad \mathrm{strongPartialRefinement}(\mathrm{pl},\ \mathrm{pl2},\ s)) \\ \quad\quad \text{WHERE } \mathrm{fm1} = F(\mathrm{pl}),\ \mathrm{pl2} = (\#F := \mathrm{fm2},\ A := A(\mathrm{pl}),\ K := K(\mathrm{pl})\#) \end{array}$$

`fmCompStrongDef`:

$$\begin{array}{l} \hline \{1\} \quad \text{FORALL } (\mathrm{pl},\ \mathrm{fm2},\ s): \\ \quad (\mathrm{fmPartialRefinement}(F(\mathrm{pl}),\ \mathrm{fm2},\ s) \wedge \mathrm{wfPL}(\mathrm{pl2}) \Rightarrow \\ \quad\quad \mathrm{strongPartialRefinement}(\mathrm{pl},\ \mathrm{pl2},\ s)) \\ \quad\quad \text{WHERE } \mathrm{fm1} = F(\mathrm{pl}),\ \mathrm{pl2} = (\#F := \mathrm{fm2},\ A := A(\mathrm{pl}),\ K := K(\mathrm{pl})\#) \end{array}$$

Expanding the definition(s) of (fmPartialRefinement strongPartialRefinement subset),

`fmCompStrongDef`:

$$\begin{array}{l} \hline \{1\} \quad \text{FORALL } (\mathrm{pl},\ \mathrm{fm2},\ s): \\ \quad ((\text{FORALL } (c\colon \mathrm{Conf}): s(c) \Rightarrow \{\text{——}\}(F(\mathrm{pl}))(c) \wedge \{\text{——}\}(\mathrm{fm2})(c)) \wedge \\ \quad\quad \mathrm{wfPL}((\#F := \mathrm{fm2},\ A := A(\mathrm{pl}),\ K := K(\mathrm{pl})\#)) \\ \quad\quad \Rightarrow \\ \quad\quad (s \subseteq \{\text{——}\}(F(\mathrm{pl}))) \wedge \\ \quad\quad\ (s \subseteq \{\text{——}\}(\mathrm{fm2})) \wedge \\ \quad\quad\ \ (\text{FORALL } c: \\ \quad\quad\quad\quad s(c) \Rightarrow \\ \quad\quad\quad\quad\ (\mathrm{prod}(\mathrm{pl},\ c) \text{ —- } \mathrm{prod}((\#F := \mathrm{fm2},\ A := A(\mathrm{pl}),\ K := K(\mathrm{pl})\#),\ c) \end{array}$$

Expanding the definition of subset?,

`fmCompStrongDef`:

$$\begin{array}{l} \hline \{1\} \quad \text{FORALL } (\mathrm{pl},\ \mathrm{fm2},\ s): \\ \quad ((\text{FORALL } (c\colon \mathrm{Conf}): s(c) \Rightarrow \{\text{——}\}(F(\mathrm{pl}))(c) \wedge \{\text{——}\}(\mathrm{fm2})(c)) \wedge \\ \quad\quad \mathrm{wfPL}((\#F := \mathrm{fm2},\ A := A(\mathrm{pl}),\ K := K(\mathrm{pl})\#)) \\ \quad\quad \Rightarrow \\ \quad\quad (\text{FORALL } (x\colon \mathrm{Conf}): (x \in s) \Rightarrow (x \in \{\text{——}\}(F(\mathrm{pl})))) \wedge \\ \quad\quad\ (\text{FORALL } (x\colon \mathrm{Conf}): (x \in s) \Rightarrow (x \in \{\text{——}\}(\mathrm{fm2}))) \wedge \\ \quad\quad\ \ (\text{FORALL } c: \\ \quad\quad\quad\quad s(c) \Rightarrow \\ \quad\quad\quad\quad\ (\mathrm{prod}(\mathrm{pl},\ c) \text{ —- } \mathrm{prod}((\#F := \mathrm{fm2},\ A := A(\mathrm{pl}),\ K := K(\mathrm{pl})\#),\ c) \end{array}$$

Expanding the definition of member,

`fmCompStrongDef`:

$$
\begin{array}{l|l}
\hline
\{1\} & \textsc{forall}\ (\mathrm{pl},\ \mathrm{fm2},\ s):\\
& \quad ((\textsc{forall}\ (c\colon \mathrm{Conf}):\ s(c) \Rightarrow \{\text{——}\}(F(\mathrm{pl}))(c) \wedge \{\text{——}\}(\mathrm{fm2})(c)) \wedge\\
& \qquad \mathrm{wfPL}((\#F := \mathrm{fm2},\ A := A(\mathrm{pl}),\ K := K(\mathrm{pl})\#))\\
& \qquad \Rightarrow\\
& \qquad (\textsc{forall}\ (x\colon \mathrm{Conf}):\ s(x) \Rightarrow \{\text{——}\}(F(\mathrm{pl}))(x)) \wedge\\
& \qquad\ (\textsc{forall}\ (x\colon \mathrm{Conf}):\ s(x) \Rightarrow \{\text{——}\}(\mathrm{fm2})(x)) \wedge\\
& \qquad\ \ (\textsc{forall}\ c:\\
& \qquad\qquad s(c) \Rightarrow\\
& \qquad\qquad\ (\mathrm{prod}(\mathrm{pl},\ c)\ \text{—-}\ \mathrm{prod}((\#F := \mathrm{fm2},\ A := A(\mathrm{pl}),\ K := K(\mathrm{pl})\#),\ c)
\end{array}
$$

Expanding the definition of prod,
`fmCompStrongDef`:

$$
\begin{array}{l|l}
\hline
\{1\} & \textsc{forall}\ (\mathrm{pl},\ \mathrm{fm2},\ s):\\
& \quad ((\textsc{forall}\ (c\colon \mathrm{Conf}):\ s(c) \Rightarrow \{\text{——}\}(F(\mathrm{pl}))(c) \wedge \{\text{——}\}(\mathrm{fm2})(c)) \wedge\\
& \qquad \mathrm{wfPL}((\#F := \mathrm{fm2},\ A := A(\mathrm{pl}),\ K := K(\mathrm{pl})\#))\\
& \qquad \Rightarrow\\
& \qquad (\textsc{forall}\ (x\colon \mathrm{Conf}):\ s(x) \Rightarrow \{\text{——}\}(F(\mathrm{pl}))(x)) \wedge\\
& \qquad\ (\textsc{forall}\ (x\colon \mathrm{Conf}):\ s(x) \Rightarrow \{\text{——}\}(\mathrm{fm2})(x)) \wedge\\
& \qquad\ \ (\textsc{forall}\ c:\\
& \qquad\qquad s(c) \Rightarrow\\
& \qquad\qquad\ (([\text{——}](K(\mathrm{pl}))(A(\mathrm{pl}))(c))\ \text{—-}\ ([\text{——}](K(\mathrm{pl}))(A(\mathrm{pl}))(c)))))
\end{array}
$$

Using lemma assetRefinement,
`fmCompStrongDef`:

$$
\begin{array}{l|l}
\{\text{-1}\} & \mathrm{orders}[\mathrm{set}[\mathrm{Asset}]].\mathrm{preorder?}(\text{—-})\\
\hline
\{1\} & \textsc{forall}\ (\mathrm{pl},\ \mathrm{fm2},\ s):\\
& \quad ((\textsc{forall}\ (c\colon \mathrm{Conf}):\ s(c) \Rightarrow \{\text{——}\}(F(\mathrm{pl}))(c) \wedge \{\text{——}\}(\mathrm{fm2})(c)) \wedge\\
& \qquad \mathrm{wfPL}((\#F := \mathrm{fm2},\ A := A(\mathrm{pl}),\ K := K(\mathrm{pl})\#))\\
& \qquad \Rightarrow\\
& \qquad (\textsc{forall}\ (x\colon \mathrm{Conf}):\ s(x) \Rightarrow \{\text{——}\}(F(\mathrm{pl}))(x)) \wedge\\
& \qquad\ (\textsc{forall}\ (x\colon \mathrm{Conf}):\ s(x) \Rightarrow \{\text{——}\}(\mathrm{fm2})(x)) \wedge\\
& \qquad\ \ (\textsc{forall}\ c:\\
& \qquad\qquad s(c) \Rightarrow\\
& \qquad\qquad\ (([\text{——}](K(\mathrm{pl}))(A(\mathrm{pl}))(c))\ \text{—-}\ ([\text{——}](K(\mathrm{pl}))(A(\mathrm{pl}))(c)))))
\end{array}
$$

Expanding the definition of preorder?,

fmCompStrongDef:

```
{-1}  reflexive?(—-) & transitive?(—-)
  |-------
{1}   FORALL (pl, fm2, s):
        ((FORALL (c: Conf): s(c) ⇒ {——}(F(pl))(c) ∧ {——}(fm2)(c)) ∧
           wfPL((#F := fm2, A := A(pl), K := K(pl)#))
           ⇒
           (FORALL (x: Conf): s(x) ⇒ {——}(F(pl))(x)) ∧
            (FORALL (x: Conf): s(x) ⇒ {——}(fm2)(x)) ∧
             (FORALL c:
                 s(c) ⇒
                  (([——](K(pl))(A(pl))(c)) —- ([——](K(pl))(A(pl))(c)))))
```

Applying disjunctive simplification to flatten sequent,
fmCompStrongDef:

```
{-1}  reflexive?(—-)
{-2}  transitive?(—-)
  |-------
{1}   FORALL (pl, fm2, s):
        ((FORALL (c: Conf): s(c) ⇒ {——}(F(pl))(c) ∧ {——}(fm2)(c)) ∧
           wfPL((#F := fm2, A := A(pl), K := K(pl)#))
           ⇒
           (FORALL (x: Conf): s(x) ⇒ {——}(F(pl))(x)) ∧
            (FORALL (x: Conf): s(x) ⇒ {——}(fm2)(x)) ∧
             (FORALL c:
                 s(c) ⇒
                  (([——](K(pl))(A(pl))(c)) —- ([——](K(pl))(A(pl))(c)))))
```

Expanding the definition of reflexive?,
fmCompStrongDef:

```
{-1}  FORALL (x: set[Asset]): (x —- x)
{-2}  transitive?(—-)
  |-------
{1}   FORALL (pl, fm2, s):
        ((FORALL (c: Conf): s(c) ⇒ {——}(F(pl))(c) ∧ {——}(fm2)(c)) ∧
           wfPL((#F := fm2, A := A(pl), K := K(pl)#))
           ⇒
           (FORALL (x: Conf): s(x) ⇒ {——}(F(pl))(x)) ∧
            (FORALL (x: Conf): s(x) ⇒ {——}(fm2)(x)) ∧
             (FORALL c:
                 s(c) ⇒
                  (([——](K(pl))(A(pl))(c)) —- ([——](K(pl))(A(pl))(c)))))
```

For the top quantifier in 1, we introduce Skolem constants: (pl fm2 s), `fmCompStrongDef`:

$$
\begin{array}{|ll}
\{-1\} & \text{FORALL } (x\text{: set}[\text{Asset}])\text{: } (x \text{ —- } x) \\
\{-2\} & \text{transitive?}(\text{—-}) \\
\hline
\{1\} & ((\text{FORALL } (c\text{: Conf})\text{: } s(c) \Rightarrow \{\text{——}\}(F(\text{pl}))(c) \wedge \{\text{——}\}(\text{fm2})(c)) \wedge \\
& \quad \text{wfPL}((\#F := \text{fm2}, A := A(\text{pl}), K := K(\text{pl})\#)) \\
& \quad \Rightarrow \\
& \quad (\text{FORALL } (x\text{: Conf})\text{: } s(x) \Rightarrow \{\text{——}\}(F(\text{pl}))(x)) \wedge \\
& \quad\; (\text{FORALL } (x\text{: Conf})\text{: } s(x) \Rightarrow \{\text{——}\}(\text{fm2})(x)) \wedge \\
& \quad\;\; (\text{FORALL } c\text{:} \\
& \qquad\quad s(c) \Rightarrow \\
& \qquad\quad\; (([\text{——}](K(\text{pl}))(A(\text{pl}))(c)) \text{ —- } ([\text{——}](K(\text{pl}))(A(\text{pl}))(c)))))
\end{array}
$$

Applying bddsimp,
we get 3 subgoals:
`fmCompStrongDef.1`:

$$
\begin{array}{|ll}
\{-1\} & \text{FORALL } (x\text{: set}[\text{Asset}])\text{: } (x \text{ —- } x) \\
\{-2\} & \text{transitive?}(\text{—-}) \\
\{-3\} & \text{FORALL } (c\text{: Conf})\text{: } s(c) \Rightarrow \{\text{——}\}(F(\text{pl}))(c) \wedge \{\text{——}\}(\text{fm2})(c) \\
\{-4\} & \text{wfPL}((\#F := \text{fm2}, A := A(\text{pl}), K := K(\text{pl})\#)) \\
\hline
\{1\} & \text{FORALL } c\text{:} \\
& \quad s(c) \Rightarrow (([\text{——}](K(\text{pl}))(A(\text{pl}))(c)) \text{ —- } ([\text{——}](K(\text{pl}))(A(\text{pl}))(c)))
\end{array}
$$

For the top quantifier in 1, we introduce Skolem constants: c,
`fmCompStrongDef.1`:

$$
\begin{array}{|ll}
\{-1\} & \text{FORALL } (x\text{: set}[\text{Asset}])\text{: } (x \text{ —- } x) \\
\{-2\} & \text{transitive?}(\text{—-}) \\
\{-3\} & \text{FORALL } (c\text{: Conf})\text{: } s(c) \Rightarrow \{\text{——}\}(F(\text{pl}))(c) \wedge \{\text{——}\}(\text{fm2})(c) \\
\{-4\} & \text{wfPL}((\#F := \text{fm2}, A := A(\text{pl}), K := K(\text{pl})\#)) \\
\hline
\{1\} & s(c) \Rightarrow (([\text{——}](K(\text{pl}))(A(\text{pl}))(c)) \text{ —- } ([\text{——}](K(\text{pl}))(A(\text{pl}))(c)))
\end{array}
$$

Instantiating the top quantifier in -1 with the terms: $([\text{——}](K(\text{pl}))(A(\text{pl}))(c))$,
`fmCompStrongDef.1`:

$$
\begin{array}{|ll}
\{-1\} & (([\text{——}](K(\text{pl}))(A(\text{pl}))(c)) \text{ —- } ([\text{——}](K(\text{pl}))(A(\text{pl}))(c))) \\
\{-2\} & \text{transitive?}(\text{—-}) \\
\{-3\} & \text{FORALL } (c\text{: Conf})\text{: } s(c) \Rightarrow \{\text{——}\}(F(\text{pl}))(c) \wedge \{\text{——}\}(\text{fm2})(c) \\
\{-4\} & \text{wfPL}((\#F := \text{fm2}, A := A(\text{pl}), K := K(\text{pl})\#)) \\
\hline
\{1\} & s(c) \Rightarrow (([\text{——}](K(\text{pl}))(A(\text{pl}))(c)) \text{ —- } ([\text{——}](K(\text{pl}))(A(\text{pl}))(c)))
\end{array}
$$

Applying bddsimp,
This completes the proof of `fmCompStrongDef.1`.

`fmCompStrongDef.2`:

{-1} FORALL (x: set[Asset]): (x —- x)
{-2} transitive?(—-)
{-3} FORALL (c: Conf): $s(c) \Rightarrow$ {——}(F(pl))(c) $\wedge$ {——}(fm2)(c)
{-4} wfPL((#F := fm2, A := A(pl), K := K(pl)#))

{1} FORALL (x: Conf): $s(x) \Rightarrow$ {——}(fm2)(x)

For the top quantifier in 1, we introduce Skolem constants: c,
`fmCompStrongDef.2`:

{-1} FORALL (x: set[Asset]): (x —- x)
{-2} transitive?(—-)
{-3} FORALL (c: Conf): $s(c) \Rightarrow$ {——}(F(pl))(c) $\wedge$ {——}(fm2)(c)
{-4} wfPL((#F := fm2, A := A(pl), K := K(pl)#))

{1} $s(c) \Rightarrow$ {——}(fm2)(c)

Instantiating the top quantifier in -3 with the terms: c,
`fmCompStrongDef.2`:

{-1} FORALL (x: set[Asset]): (x —- x)
{-2} transitive?(—-)
{-3} $s(c) \Rightarrow$ {——}(F(pl))(c) $\wedge$ {——}(fm2)(c)
{-4} wfPL((#F := fm2, A := A(pl), K := K(pl)#))

{1} $s(c) \Rightarrow$ {——}(fm2)(c)

Applying bddsimp,
This completes the proof of `fmCompStrongDef.2`.
`fmCompStrongDef.3`:

{-1} FORALL (x: set[Asset]): (x —- x)
{-2} transitive?(—-)
{-3} FORALL (c: Conf): $s(c) \Rightarrow$ {——}(F(pl))(c) $\wedge$ {——}(fm2)(c)
{-4} wfPL((#F := fm2, A := A(pl), K := K(pl)#))

{1} FORALL (x: Conf): $s(x) \Rightarrow$ {——}(F(pl))(x)

For the top quantifier in 1, we introduce Skolem constants: c,
`fmCompStrongDef.3`:

{-1} FORALL (x: set[Asset]): (x —- x)
{-2} transitive?(—-)
{-3} FORALL (c: Conf): $s(c) \Rightarrow$ {——}(F(pl))(c) $\wedge$ {——}(fm2)(c)
{-4} wfPL((#F := fm2, A := A(pl), K := K(pl)#))

{1} $s(c) \Rightarrow$ {——}(F(pl))(c)

Instantiating the top quantifier in -3 with the terms: c,

`fmCompStrongDef.3`:

{-1}	FORALL (x: set[Asset]): (x —- x)
{-2}	transitive?(—-)
{-3}	$s(c) \Rightarrow$ {——}(F(pl))(c) $\wedge$ {——}(fm2)(c)
{-4}	wfPL((#F := fm2, A := A(pl), K := K(pl)#))
{1}	$s(c) \Rightarrow$ {——}(F(pl))(c)

Applying bddsimp,
This completes the proof of `fmCompStrongDef.3`.
Q.E.D.

Verbose proof for `partPlusTotalImpliesPartFun`.
`partPlusTotalImpliesPartFun`:

|-------
{1} FORALL pl1, pl2, pl3, s:
strongPartialRefinement(pl1, pl2, s) $\wedge$ plRefinement(pl2, pl3) $\Rightarrow$
(EXISTS (f: $[(s) \to (\{\text{——}\}(F(\text{pl3})))]$):
(FORALL c:
$s(c)$ $\Rightarrow$
($\{\text{——}\}(F(\text{pl3}))(f(c))$ $\wedge$ (prod(pl1, c) —-
prod(pl3, $f(c)$))))))

`partPlusTotalImpliesPartFun`:

|-------
{1} FORALL pl1, pl2, pl3, s:
strongPartialRefinement(pl1, pl2, s) $\wedge$ plRefinement(pl2, pl3) $\Rightarrow$
(EXISTS (f: $[(s) \to (\{\text{——}\}(F(\text{pl3})))]$):
(FORALL c:
$s(c)$ $\Rightarrow$
($\{\text{——}\}(F(\text{pl3}))(f(c))$ $\wedge$ (prod(pl1, c) —-
prod(pl3, $f(c)$))))))

For the top quantifier in 1, we introduce Skolem constants: (pl1 pl2 pl3 s),
`partPlusTotalImpliesPartFun`:

|-------
{1} strongPartialRefinement(pl1, pl2, s) $\wedge$ plRefinement(pl2, pl3) $\Rightarrow$
(EXISTS (f: $[(s) \to (\{\text{——}\}(F(\text{pl3})))]$):
FORALL c:
$s(c)$ $\Rightarrow$
($\{\text{——}\}(F(\text{pl3}))(f(c))$ $\wedge$ (prod(pl1, c) —- prod(pl3, $f(c)$)))))

Applying bddsimp,
`partPlusTotalImpliesPartFun`:

{-1} strongPartialRefinement(pl1, pl2, s)
{-2} plRefinement(pl2, pl3)
|-------
{1} EXISTS (f: $[(s) \to (\{\text{——}\}(F(\text{pl3})))]$):
FORALL c:
$s(c)$ $\Rightarrow$ ($\{\text{——}\}(F(\text{pl3}))(f(c))$ $\wedge$ (prod(pl1, c) —- prod(pl3, $f(c)$))))

Applying totalRefIFFExistsFun

`partPlusTotalImpliesPartFun:`

{-1} ∀ (pl1: PL[Conf, FM, Asset, AssetName, CK, {——}, [——]],
pl2: PL[Conf, FM, Asset, AssetName, CK, {——}, [——]]):
plRefinement(pl1, pl2) ⇔
(EXISTS (f: [({——}(F(pl1))) → ({——}(F(pl2)))]):
plRefinementFun(pl1, pl2, f))
{-2} strongPartialRefinement(pl1, pl2, s)
{-3} plRefinement(pl2, pl3)

{1} EXISTS (f: [(s) → ({——}(F(pl3)))]):
FORALL c:
$s(c)$ ⇒ ({——}(F(pl3))($f(c)$) ∧ (prod(pl1, c) —- prod(pl3, $f(c)$)))

Applying partRefExistsFunId

`partPlusTotalImpliesPartFun:`

{-1} ∀ (pl1, pl2, s):
strongPartialRefinement(pl1, pl2, s) ⇒
(EXISTS (f: [(s) → (s)]):
(FORALL c:
$s(c)$ ⇒
({——}(F(pl2))($f(c)$)) ∧ (prod(pl1, c) —-
prod(pl2, $f(c)$)))))
{-2} ∀ (pl1: PL[Conf, FM, Asset, AssetName, CK, {——}, [——]],
pl2: PL[Conf, FM, Asset, AssetName, CK, {——}, [——]]):
plRefinement(pl1, pl2) ⇔
(EXISTS (f: [({——}(F(pl1))) → ({——}(F(pl2)))]):
plRefinementFun(pl1, pl2, f))
{-3} strongPartialRefinement(pl1, pl2, s)
{-4} plRefinement(pl2, pl3)

{1} EXISTS (f: [(s) → ({——}(F(pl3)))]):
FORALL c:
$s(c)$ ⇒ ({——}(F(pl3))($f(c)$) ∧ (prod(pl1, c) —- prod(pl3, $f(c)$)))

Instantiating the top quantifier in -1 with the terms: pl1, pl2, s,

`partPlusTotalImpliesPartFun`:

{-1} strongPartialRefinement(pl1, pl2, s) ⇒
(EXISTS (f: [(s) → (s)]):
FORALL c:
$s(c)$ ⇒
({——}(F(pl2))($f(c)$)) ∧ (prod(pl1, c) —- prod(pl2, $f(c)$)))

{-2} ∀ (pl1: PL[Conf, FM, Asset, AssetName, CK, {——}, [——]],
pl2: PL[Conf, FM, Asset, AssetName, CK, {——}, [——]]):
plRefinement(pl1, pl2) ⇔
(EXISTS (f: [({——}(F(pl1))) → ({——}(F(pl2)))]):
plRefinementFun(pl1, pl2, f))

{-3} strongPartialRefinement(pl1, pl2, s)

{-4} plRefinement(pl2, pl3)

{1} EXISTS (f: [(s) → ({——}(F(pl3)))]):
FORALL c:
$s(c)$ ⇒ ({——}(F(pl3))($f(c)$) ∧ (prod(pl1, c) —- prod(pl3, $f(c)$)))

Applying bddsimp,

`partPlusTotalImpliesPartFun`:

{-1} strongPartialRefinement(pl1, pl2, s)

{-2} EXISTS (f: [(s) → (s)]):
FORALL c:
$s(c)$ ⇒ ({——}(F(pl2))($f(c)$)) ∧ (prod(pl1, c) —- prod(pl2, $f(c)$))

{-3} ∀ (pl1: PL[Conf, FM, Asset, AssetName, CK, {——}, [——]],
pl2: PL[Conf, FM, Asset, AssetName, CK, {——}, [——]]):
plRefinement(pl1, pl2) ⇔
(EXISTS (f: [({——}(F(pl1))) → ({——}(F(pl2)))]):
plRefinementFun(pl1, pl2, f))

{-4} plRefinement(pl2, pl3)

{1} EXISTS (f: [(s) → ({——}(F(pl3)))]):
FORALL c:
$s(c)$ ⇒ ({——}(F(pl3))($f(c)$) ∧ (prod(pl1, c) —- prod(pl3, $f(c)$)))

For the top quantifier in -2, we introduce Skolem constants: f,

`partPlusTotalImpliesPartFun`:

$$
\begin{array}{|ll}
\{\text{-1}\} & \text{strongPartialRefinement}(\text{pl1},\ \text{pl2},\ s) \\
\{\text{-2}\} & \text{FORALL } c: \\
 & \quad s(c) \Rightarrow (\{\text{——}\}(F(\text{pl2}))(f(c))) \wedge (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl2},\ f(c))) \\
\{\text{-3}\} & \forall\ (\text{pl1: PL}[\text{Conf, FM, Asset, AssetName, CK, }\{\text{——}\},\ [\text{——}]], \\
 & \qquad \text{pl2: PL}[\text{Conf, FM, Asset, AssetName, CK, }\{\text{——}\},\ [\text{——}]]): \\
 & \quad \text{plRefinement}(\text{pl1},\ \text{pl2}) \Leftrightarrow \\
 & \quad\ (\text{EXISTS } (f\text{: } [(\{\text{——}\}(F(\text{pl1}))) \to (\{\text{——}\}(F(\text{pl2})))]): \\
 & \qquad \text{plRefinementFun}(\text{pl1},\ \text{pl2},\ f)) \\
\{\text{-4}\} & \text{plRefinement}(\text{pl2},\ \text{pl3}) \\
\hline
\{1\} & \text{EXISTS } (f\text{: } [(s) \to (\{\text{——}\}(F(\text{pl3})))]): \\
 & \quad \text{FORALL } c: \\
 & \qquad s(c) \Rightarrow (\{\text{——}\}(F(\text{pl3}))(f(c)) \wedge (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl3},\ f(c))))
\end{array}
$$

Instantiating the top quantifier in -3 with the terms: pl2, pl3,
`partPlusTotalImpliesPartFun`:

$$
\begin{array}{|ll}
\{\text{-1}\} & \text{strongPartialRefinement}(\text{pl1},\ \text{pl2},\ s) \\
\{\text{-2}\} & \text{FORALL } c: \\
 & \quad s(c) \Rightarrow (\{\text{——}\}(F(\text{pl2}))(f(c))) \wedge (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl2},\ f(c))) \\
\{\text{-3}\} & \text{plRefinement}(\text{pl2},\ \text{pl3}) \Leftrightarrow \\
 & \ (\text{EXISTS } (f\text{: } [(\{\text{——}\}(F(\text{pl2}))) \to (\{\text{——}\}(F(\text{pl3})))]): \\
 & \qquad \text{plRefinementFun}(\text{pl2},\ \text{pl3},\ f)) \\
\{\text{-4}\} & \text{plRefinement}(\text{pl2},\ \text{pl3}) \\
\hline
\{1\} & \text{EXISTS } (f\text{: } [(s) \to (\{\text{——}\}(F(\text{pl3})))]): \\
 & \quad \text{FORALL } c: \\
 & \qquad s(c) \Rightarrow (\{\text{——}\}(F(\text{pl3}))(f(c)) \wedge (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl3},\ f(c))))
\end{array}
$$

Applying bddsimp,
`partPlusTotalImpliesPartFun`:

$$
\begin{array}{|ll}
\{\text{-1}\} & \text{strongPartialRefinement}(\text{pl1},\ \text{pl2},\ s) \\
\{\text{-2}\} & \text{FORALL } c: \\
 & \quad s(c) \Rightarrow (\{\text{——}\}(F(\text{pl2}))(f(c))) \wedge (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl2},\ f(c))) \\
\{\text{-3}\} & \text{plRefinement}(\text{pl2},\ \text{pl3}) \\
\{\text{-4}\} & \text{EXISTS } (f\text{: } [(\{\text{——}\}(F(\text{pl2}))) \to (\{\text{——}\}(F(\text{pl3})))]): \text{plRefinement-} \\
 & \text{Fun}(\text{pl2},\ \text{pl3},\ f) \\
\hline
\{1\} & \text{EXISTS } (f\text{: } [(s) \to (\{\text{——}\}(F(\text{pl3})))]): \\
 & \quad \text{FORALL } c: \\
 & \qquad s(c) \Rightarrow (\{\text{——}\}(F(\text{pl3}))(f(c)) \wedge (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl3},\ f(c))))
\end{array}
$$

For the top quantifier in -4, we introduce Skolem constants: g,

`partPlusTotalImpliesPartFun`:

$$
\begin{array}{|ll}
\{\text{-}1\} & \text{strongPartialRefinement}(\text{pl1},\ \text{pl2},\ s) \\
\{\text{-}2\} & \textsc{forall}\ c: \\
& \quad s(c) \Rightarrow (\{\text{——}\}(F(\text{pl2}))(f(c))) \wedge (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl2},\ f(c))) \\
\{\text{-}3\} & \text{plRefinement}(\text{pl2},\ \text{pl3}) \\
\{\text{-}4\} & \text{plRefinementFun}(\text{pl2},\ \text{pl3},\ g) \\
\hline
\{1\} & \textsc{exists}\ (f\colon [(s) \to (\{\text{——}\}(F(\text{pl3})))]): \\
& \quad \textsc{forall}\ c: \\
& \quad\quad s(c) \Rightarrow (\{\text{——}\}(F(\text{pl3}))(f(c)) \wedge (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl3},\ f(c))))
\end{array}
$$

Instantiating the top quantifier in 1 with the terms: $g \circ f$,
we get 2 subgoals:
`partPlusTotalImpliesPartFun.1`:

$$
\begin{array}{|ll}
\{\text{-}1\} & \text{strongPartialRefinement}(\text{pl1},\ \text{pl2},\ s) \\
\{\text{-}2\} & \textsc{forall}\ c: \\
& \quad s(c) \Rightarrow (\{\text{——}\}(F(\text{pl2}))(f(c))) \wedge (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl2},\ f(c))) \\
\{\text{-}3\} & \text{plRefinement}(\text{pl2},\ \text{pl3}) \\
\{\text{-}4\} & \text{plRefinementFun}(\text{pl2},\ \text{pl3},\ g) \\
\hline
\{1\} & \textsc{forall}\ c: \\
& \quad s(c) \Rightarrow \\
& \quad\ (\{\text{——}\}(F(\text{pl3}))((g \circ f)(c)) \wedge \\
& \quad\quad (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl3},\ (g \circ f)(c))))
\end{array}
$$

For the top quantifier in 1, we introduce Skolem constants: c,
`partPlusTotalImpliesPartFun.1`:

$$
\begin{array}{|ll}
\{\text{-}1\} & \text{strongPartialRefinement}(\text{pl1},\ \text{pl2},\ s) \\
\{\text{-}2\} & \textsc{forall}\ c: \\
& \quad s(c) \Rightarrow (\{\text{——}\}(F(\text{pl2}))(f(c))) \wedge (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl2},\ f(c))) \\
\{\text{-}3\} & \text{plRefinement}(\text{pl2},\ \text{pl3}) \\
\{\text{-}4\} & \text{plRefinementFun}(\text{pl2},\ \text{pl3},\ g) \\
\hline
\{1\} & s(c) \Rightarrow \\
& \ (\{\text{——}\}(F(\text{pl3}))((g \circ f)(c)) \wedge \\
& \quad (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl3},\ (g \circ f)(c))))
\end{array}
$$

Expanding the definition of o,
`partPlusTotalImpliesPartFun.1`:

$$
\begin{array}{|ll}
\{\text{-}1\} & \text{strongPartialRefinement}(\text{pl1},\ \text{pl2},\ s) \\
\{\text{-}2\} & \textsc{forall}\ c: \\
& \quad s(c) \Rightarrow (\{\text{——}\}(F(\text{pl2}))(f(c))) \wedge (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl2},\ f(c))) \\
\{\text{-}3\} & \text{plRefinement}(\text{pl2},\ \text{pl3}) \\
\{\text{-}4\} & \text{plRefinementFun}(\text{pl2},\ \text{pl3},\ g) \\
\hline
\{1\} & s(c) \Rightarrow (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl3},\ g(f(c))))
\end{array}
$$

Instantiating the top quantifier in -2 with the terms: c,
`partPlusTotalImpliesPartFun.1:`

{-1}	strongPartialRefinement(pl1, pl2, s)
{-2}	$s(c)$ $\Rightarrow$ ({——}(F(pl2))($f(c)$)) $\wedge$ (prod(pl1, c) —- prod(pl2, $f(c)$))
{-3}	plRefinement(pl2, pl3)
{-4}	plRefinementFun(pl2, pl3, g)
{1}	$s(c)$ $\Rightarrow$ (prod(pl1, c) —- prod(pl3, $g(f(c))$))

Applying bddsimp,
`partPlusTotalImpliesPartFun.1:`

{-1}	strongPartialRefinement(pl1, pl2, s)
{-2}	$s(c)$
{-3}	({——}(F(pl2))($f(c)$))
{-4}	(prod(pl1, c) —- prod(pl2, $f(c)$))
{-5}	plRefinement(pl2, pl3)
{-6}	plRefinementFun(pl2, pl3, g)
{1}	(prod(pl1, c) —- prod(pl3, $g(f(c))$))

Expanding the definition of plRefinementFun,
`partPlusTotalImpliesPartFun.1:`

{-1}	strongPartialRefinement(pl1, pl2, s)
{-2}	$s(c)$
{-3}	({——}(F(pl2))($f(c)$))
{-4}	(prod(pl1, c) —- prod(pl2, $f(c)$))
{-5}	plRefinement(pl2, pl3)
{-6}	FORALL (c: Conf): {——}(F(pl2))(c) $\Rightarrow$ (prod(pl2, c) —- prod(pl3, $g(c)$))
{1}	(prod(pl1, c) —- prod(pl3, $g(f(c))$))

Instantiating the top quantifier in -6 with the terms: $f(c)$,
`partPlusTotalImpliesPartFun.1:`

{-1}	strongPartialRefinement(pl1, pl2, s)
{-2}	$s(c)$
{-3}	({——}(F(pl2))($f(c)$))
{-4}	(prod(pl1, c) —- prod(pl2, $f(c)$))
{-5}	plRefinement(pl2, pl3)
{-6}	{——}(F(pl2))($f(c)$) $\Rightarrow$ (prod(pl2, $f(c)$) —- prod(pl3, $g(f(c))$))
{1}	(prod(pl1, c) —- prod(pl3, $g(f(c))$))

Applying bddsimp,

`partPlusTotalImpliesPartFun.1:`

{-1}	strongPartialRefinement(pl1, pl2, s)
{-2}	$s(c)$
{-3}	({——}(F(pl2))($f(c)$))
{-4}	(prod(pl1, c) —- prod(pl2, $f(c)$))
{-5}	plRefinement(pl2, pl3)
{-6}	(prod(pl2, $f(c)$) —- prod(pl3, $g(f(c))$))
{1}	(prod(pl1, c) —- prod(pl3, $g(f(c))$))

Using lemma assetRefinement,
`partPlusTotalImpliesPartFun.1:`

{-1}	orders[set[Asset]].preorder?(—-)
{-2}	strongPartialRefinement(pl1, pl2, s)
{-3}	$s(c)$
{-4}	({——}(F(pl2))($f(c)$))
{-5}	(prod(pl1, c) —- prod(pl2, $f(c)$))
{-6}	plRefinement(pl2, pl3)
{-7}	(prod(pl2, $f(c)$) —- prod(pl3, $g(f(c))$))
{1}	(prod(pl1, c) —- prod(pl3, $g(f(c))$))

Expanding the definition of preorder?,
`partPlusTotalImpliesPartFun.1:`

{-1}	reflexive?(—-) & transitive?(—-)
{-2}	strongPartialRefinement(pl1, pl2, s)
{-3}	$s(c)$
{-4}	({——}(F(pl2))($f(c)$))
{-5}	(prod(pl1, c) —- prod(pl2, $f(c)$))
{-6}	plRefinement(pl2, pl3)
{-7}	(prod(pl2, $f(c)$) —- prod(pl3, $g(f(c))$))
{1}	(prod(pl1, c) —- prod(pl3, $g(f(c))$))

Applying disjunctive simplification to flatten sequent,
`partPlusTotalImpliesPartFun.1:`

{-1}	reflexive?(—-)
{-2}	transitive?(—-)
{-3}	strongPartialRefinement(pl1, pl2, s)
{-4}	$s(c)$
{-5}	({——}(F(pl2))($f(c)$))
{-6}	(prod(pl1, c) —- prod(pl2, $f(c)$))
{-7}	plRefinement(pl2, pl3)
{-8}	(prod(pl2, $f(c)$) —- prod(pl3, $g(f(c))$))
{1}	(prod(pl1, c) —- prod(pl3, $g(f(c))$))

Expanding the definition of transitive?,
`partPlusTotalImpliesPartFun.1:`

{-1}	reflexive?(—-)
{-2}	FORALL (x: set[Asset]), (y: set[Asset]), (z: set[Asset]): (x —- y) & (y —- z) ⇒ (x —- z)
{-3}	strongPartialRefinement(pl1, pl2, s)
{-4}	$s(c)$
{-5}	({——}(F(pl2))($f(c)$))
{-6}	(prod(pl1, c) —- prod(pl2, $f(c)$))
{-7}	plRefinement(pl2, pl3)
{-8}	(prod(pl2, $f(c)$) —- prod(pl3, $g(f(c))$))
{1}	(prod(pl1, c) —- prod(pl3, $g(f(c))$))

Instantiating the top quantifier in -2 with the terms: prod(pl1, c), prod(pl2, $f(c)$), prod(pl3, $g(f(c))$),
`partPlusTotalImpliesPartFun.1:`

{-1}	reflexive?(—-)
{-2}	(prod(pl1, c) —- prod(pl2, $f(c)$)) & (prod(pl2, $f(c)$) —- prod(pl3, $g(f(c))$)) ⇒ (prod(pl1, c) —- prod(pl3, $g(f(c))$))
{-3}	strongPartialRefinement(pl1, pl2, s)
{-4}	$s(c)$
{-5}	({——}(F(pl2))($f(c)$))
{-6}	(prod(pl1, c) —- prod(pl2, $f(c)$))
{-7}	plRefinement(pl2, pl3)
{-8}	(prod(pl2, $f(c)$) —- prod(pl3, $g(f(c))$))
{1}	(prod(pl1, c) —- prod(pl3, $g(f(c))$))

Applying bddsimp,
This completes the proof of `partPlusTotalImpliesPartFun.1`.
`partPlusTotalImpliesPartFun.2:`

{-1}	strongPartialRefinement(pl1, pl2, s)
{-2}	FORALL c: $s(c)$ ⇒ ({——}(F(pl2))($f(c)$)) ∧ (prod(pl1, c) —- prod(pl2, $f(c)$))
{-3}	plRefinement(pl2, pl3)
{-4}	plRefinementFun(pl2, pl3, g)
{1}	∀ (x_1: (s)): {——}(F(pl2))($f(x_1)$)

For the top quantifier in 1, we introduce Skolem constants: c,

`partPlusTotalImpliesPartFun.2:`

$$
\begin{array}{r|l}
\{-1\} & \text{strongPartialRefinement}(\text{pl1},\ \text{pl2},\ s) \\
\{-2\} & \text{FORALL } c: \\
 & \quad s(c) \Rightarrow (\{\text{——}\}(F(\text{pl2}))(f(c))) \wedge (\text{prod}(\text{pl1},\ c) \text{—-} \text{prod}(\text{pl2},\ f(c))) \\
\{-3\} & \text{plRefinement}(\text{pl2},\ \text{pl3}) \\
\{-4\} & \text{plRefinementFun}(\text{pl2},\ \text{pl3},\ g) \\
\hline
\{1\} & \{\text{——}\}(F(\text{pl2}))(f(c))
\end{array}
$$

Expanding the definition of strongPartialRefinement,
`partPlusTotalImpliesPartFun.2:`

$$
\begin{array}{r|l}
\{-1\} & (s \subseteq \{\text{——}\}(F(\text{pl1}))) \wedge \\
 & \quad (s \subseteq \{\text{——}\}(F(\text{pl2}))) \wedge (\text{FORALL } c:\ s(c) \Rightarrow (\text{prod}(\text{pl1},\ c) \text{—-} \\
 & \text{prod}(\text{pl2},\ c))) \\
\{-2\} & \text{FORALL } c: \\
 & \quad s(c) \Rightarrow (\{\text{——}\}(F(\text{pl2}))(f(c))) \wedge (\text{prod}(\text{pl1},\ c) \text{—-} \text{prod}(\text{pl2},\ f(c))) \\
\{-3\} & \text{plRefinement}(\text{pl2},\ \text{pl3}) \\
\{-4\} & \text{plRefinementFun}(\text{pl2},\ \text{pl3},\ g) \\
\hline
\{1\} & \{\text{——}\}(F(\text{pl2}))(f(c))
\end{array}
$$

Applying disjunctive simplification to flatten sequent,
`partPlusTotalImpliesPartFun.2:`

$$
\begin{array}{r|l}
\{-1\} & (s \subseteq \{\text{——}\}(F(\text{pl1}))) \\
\{-2\} & (s \subseteq \{\text{——}\}(F(\text{pl2}))) \\
\{-3\} & \text{FORALL } c:\ s(c) \Rightarrow (\text{prod}(\text{pl1},\ c) \text{—-} \text{prod}(\text{pl2},\ c)) \\
\{-4\} & \text{FORALL } c: \\
 & \quad s(c) \Rightarrow (\{\text{——}\}(F(\text{pl2}))(f(c))) \wedge (\text{prod}(\text{pl1},\ c) \text{—-} \text{prod}(\text{pl2},\ f(c))) \\
\{-5\} & \text{plRefinement}(\text{pl2},\ \text{pl3}) \\
\{-6\} & \text{plRefinementFun}(\text{pl2},\ \text{pl3},\ g) \\
\hline
\{1\} & \{\text{——}\}(F(\text{pl2}))(f(c))
\end{array}
$$

Expanding the definition of subset?,
`partPlusTotalImpliesPartFun.2:`

$$
\begin{array}{r|l}
\{-1\} & \text{FORALL } (x:\ \text{Conf}):\ (x \in s) \Rightarrow (x \in \{\text{——}\}(F(\text{pl1}))) \\
\{-2\} & \text{FORALL } (x:\ \text{Conf}):\ (x \in s) \Rightarrow (x \in \{\text{——}\}(F(\text{pl2}))) \\
\{-3\} & \text{FORALL } c:\ s(c) \Rightarrow (\text{prod}(\text{pl1},\ c) \text{—-} \text{prod}(\text{pl2},\ c)) \\
\{-4\} & \text{FORALL } c: \\
 & \quad s(c) \Rightarrow (\{\text{——}\}(F(\text{pl2}))(f(c))) \wedge (\text{prod}(\text{pl1},\ c) \text{—-} \text{prod}(\text{pl2},\ f(c))) \\
\{-5\} & \text{plRefinement}(\text{pl2},\ \text{pl3}) \\
\{-6\} & \text{plRefinementFun}(\text{pl2},\ \text{pl3},\ g) \\
\hline
\{1\} & \{\text{——}\}(F(\text{pl2}))(f(c))
\end{array}
$$

Instantiating the top quantifier in -2 with the terms: c,

`partPlusTotalImpliesPartFun.2:`

{-1}	FORALL $(x\colon \text{Conf})\colon\ (x \in s) \Rightarrow (x \in \{\text{——}\}(F(\text{pl1})))$
{-2}	$(c \in s) \Rightarrow (c \in \{\text{——}\}(F(\text{pl2})))$
{-3}	FORALL $c\colon\ s(c) \Rightarrow (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl2},\ c))$
{-4}	FORALL c: $s(c) \Rightarrow (\{\text{——}\}(F(\text{pl2}))(f(c))) \wedge (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl2},\ f(c)))$
{-5}	plRefinement(pl2, pl3)
{-6}	plRefinementFun(pl2, pl3, g)
{1}	$\{\text{——}\}(F(\text{pl2}))(f(c))$

Expanding the definition of member,
`partPlusTotalImpliesPartFun.2:`

{-1}	FORALL $(x\colon \text{Conf})\colon\ s(x) \Rightarrow \{\text{——}\}(F(\text{pl1}))(x)$
{-2}	$\{\text{——}\}(F(\text{pl2}))(c)$
{-3}	FORALL $c\colon\ s(c) \Rightarrow (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl2},\ c))$
{-4}	FORALL c: $s(c) \Rightarrow (\{\text{——}\}(F(\text{pl2}))(f(c))) \wedge (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl2},\ f(c)))$
{-5}	plRefinement(pl2, pl3)
{-6}	plRefinementFun(pl2, pl3, g)
{1}	$\{\text{——}\}(F(\text{pl2}))(f(c))$

Instantiating the top quantifier in -4 with the terms: c,
`partPlusTotalImpliesPartFun.2:`

{-1}	FORALL $(x\colon \text{Conf})\colon\ s(x) \Rightarrow \{\text{——}\}(F(\text{pl1}))(x)$
{-2}	$\{\text{——}\}(F(\text{pl2}))(c)$
{-3}	FORALL $c\colon\ s(c) \Rightarrow (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl2},\ c))$
{-4}	$s(c) \Rightarrow (\{\text{——}\}(F(\text{pl2}))(f(c))) \wedge (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl2},\ f(c)))$
{-5}	plRefinement(pl2, pl3)
{-6}	plRefinementFun(pl2, pl3, g)
{1}	$\{\text{——}\}(F(\text{pl2}))(f(c))$

Applying bddsimp,
`partPlusTotalImpliesPartFun.2:`

{-1}	FORALL $(x\colon \text{Conf})\colon\ s(x) \Rightarrow \{\text{——}\}(F(\text{pl1}))(x)$
{-2}	$\{\text{——}\}(F(\text{pl2}))(c)$
{-3}	FORALL $c\colon\ s(c) \Rightarrow (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl2},\ c))$
{-4}	plRefinement(pl2, pl3)
{-5}	plRefinementFun(pl2, pl3, g)
{1}	$\{\text{——}\}(F(\text{pl2}))(f(c))$
{2}	$s(c)$

Adding type constraints for c,

`partPlusTotalImpliesPartFun.2`:

{-1}	$s(c)$
{-2}	FORALL (x: Conf): $s(x) \Rightarrow$ {——}(F(pl1))(x)
{-3}	{——}(F(pl2))(c)
{-4}	FORALL c: $s(c) \Rightarrow$ (prod(pl1, c) —- prod(pl2, c))
{-5}	plRefinement(pl2, pl3)
{-6}	plRefinementFun(pl2, pl3, g)
{1}	{——}(F(pl2))($f(c)$)
{2}	$s(c)$

which is trivially true.
This completes the proof of `partPlusTotalImpliesPartFun.2`.
Q.E.D.

Verbose proof for `partPlusTotalStrongerImpliesPart`.
`partPlusTotalStrongerImpliesPart:`

|-------
{1} FORALL pl1, pl2, pl3, s:
strongPartialRefinement(pl1, pl2, s) $\wedge$ strongerPLrefinement(pl2, pl3) $\Rightarrow$
strongPartialRefinement(pl1, pl3, s)

`partPlusTotalStrongerImpliesPart:`

|-------
{1} FORALL pl1, pl2, pl3, s:
strongPartialRefinement(pl1, pl2, s) $\wedge$ strongerPLrefinement(pl2, pl3) $\Rightarrow$
strongPartialRefinement(pl1, pl3, s)

For the top quantifier in 1, we introduce Skolem constants: (pl1 pl2 pl3 s),
`partPlusTotalStrongerImpliesPart:`

|-------
{1} strongPartialRefinement(pl1, pl2, s) $\wedge$ strongerPLrefinement(pl2, pl3) $\Rightarrow$
strongPartialRefinement(pl1, pl3, s)

Applying bddsimp,
`partPlusTotalStrongerImpliesPart:`

{-1} strongPartialRefinement(pl1, pl2, s)
{-2} strongerPLrefinement(pl2, pl3)
|-------
{1} strongPartialRefinement(pl1, pl3, s)

Expanding the definition(s) of (strongPartialRefinement strongerPLrefinement),
`partPlusTotalStrongerImpliesPart:`

{-1} $(s \subseteq \{\text{——}\}(F(\text{pl1}))) \wedge$
$(s \subseteq \{\text{——}\}(F(\text{pl2}))) \wedge$ (FORALL c: $s(c) \Rightarrow$ (prod(pl1, c) —- prod(pl2, c)))
{-2} FORALL (c_1: Conf):
$\{\text{——}\}(F(\text{pl2}))(c_1) \Rightarrow$
$(\{\text{——}\}(F(\text{pl3}))(c_1) \wedge$
$(([\text{——}](K(\text{pl2}))(A(\text{pl2}))(c_1))$ —- $([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c_1))))$
|-------
{1} $(s \subseteq \{\text{——}\}(F(\text{pl1}))) \wedge$
$(s \subseteq \{\text{——}\}(F(\text{pl3}))) \wedge$ (FORALL c: $s(c) \Rightarrow$ (prod(pl1, c) —- prod(pl3, c)))

Expanding the definition of subset?,

`partPlusTotalStrongerImpliesPart:`

$$
\begin{array}{l|l}
\{\text{-1}\} & (\text{FORALL } (x\text{: Conf}): (x \in s) \Rightarrow (x \in \{\text{——}\}(F(\text{pl1})))) \wedge \\
 & \quad (\text{FORALL } (x\text{: Conf}): (x \in s) \Rightarrow (x \in \{\text{——}\}(F(\text{pl2})))) \wedge \\
 & \quad\; (\text{FORALL } c\text{: } s(c) \Rightarrow (\text{prod}(\text{pl1}, c) \text{ —- } \text{prod}(\text{pl2}, c))) \\
\{\text{-2}\} & \text{FORALL } (c_1\text{: Conf}): \\
 & \quad \{\text{——}\}(F(\text{pl2}))(c_1) \Rightarrow \\
 & \quad\; (\{\text{——}\}(F(\text{pl3}))(c_1) \wedge \\
 & \qquad (([\text{——}](K(\text{pl2}))(A(\text{pl2}))(c_1)) \text{ —- } ([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c_1)))) \\
\hline
\{1\} & (\text{FORALL } (x\text{: Conf}): (x \in s) \Rightarrow (x \in \{\text{——}\}(F(\text{pl1})))) \wedge \\
 & \quad (\text{FORALL } (x\text{: Conf}): (x \in s) \Rightarrow (x \in \{\text{——}\}(F(\text{pl3})))) \wedge \\
 & \quad\; (\text{FORALL } c\text{: } s(c) \Rightarrow (\text{prod}(\text{pl1}, c) \text{ —- } \text{prod}(\text{pl3}, c)))
\end{array}
$$

Expanding the definition of member,
`partPlusTotalStrongerImpliesPart:`

$$
\begin{array}{l|l}
\{\text{-1}\} & (\text{FORALL } (x\text{: Conf}): s(x) \Rightarrow \{\text{——}\}(F(\text{pl1}))(x)) \wedge \\
 & \quad (\text{FORALL } (x\text{: Conf}): s(x) \Rightarrow \{\text{——}\}(F(\text{pl2}))(x)) \wedge \\
 & \quad\; (\text{FORALL } c\text{: } s(c) \Rightarrow (\text{prod}(\text{pl1}, c) \text{ —- } \text{prod}(\text{pl2}, c))) \\
\{\text{-2}\} & \text{FORALL } (c_1\text{: Conf}): \\
 & \quad \{\text{——}\}(F(\text{pl2}))(c_1) \Rightarrow \\
 & \quad\; (\{\text{——}\}(F(\text{pl3}))(c_1) \wedge \\
 & \qquad (([\text{——}](K(\text{pl2}))(A(\text{pl2}))(c_1)) \text{ —- } ([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c_1)))) \\
\hline
\{1\} & (\text{FORALL } (x\text{: Conf}): s(x) \Rightarrow \{\text{——}\}(F(\text{pl1}))(x)) \wedge \\
 & \quad (\text{FORALL } (x\text{: Conf}): s(x) \Rightarrow \{\text{——}\}(F(\text{pl3}))(x)) \wedge \\
 & \quad\; (\text{FORALL } c\text{: } s(c) \Rightarrow (\text{prod}(\text{pl1}, c) \text{ —- } \text{prod}(\text{pl3}, c)))
\end{array}
$$

Expanding the definition of prod,
`partPlusTotalStrongerImpliesPart:`

$$
\begin{array}{l|l}
\{\text{-1}\} & (\text{FORALL } (x\text{: Conf}): s(x) \Rightarrow \{\text{——}\}(F(\text{pl1}))(x)) \wedge \\
 & \quad (\text{FORALL } (x\text{: Conf}): s(x) \Rightarrow \{\text{——}\}(F(\text{pl2}))(x)) \wedge \\
 & \quad\; (\text{FORALL } c\text{:} \\
 & \qquad s(c) \Rightarrow \\
 & \qquad\; (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c)) \text{ —- } ([\text{——}](K(\text{pl2}))(A(\text{pl2}))(c)))) \\
\{\text{-2}\} & \text{FORALL } (c_1\text{: Conf}): \\
 & \quad \{\text{——}\}(F(\text{pl2}))(c_1) \Rightarrow \\
 & \quad\; (\{\text{——}\}(F(\text{pl3}))(c_1) \wedge \\
 & \qquad (([\text{——}](K(\text{pl2}))(A(\text{pl2}))(c_1)) \text{ —- } ([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c_1)))) \\
\hline
\{1\} & (\text{FORALL } (x\text{: Conf}): s(x) \Rightarrow \{\text{——}\}(F(\text{pl1}))(x)) \wedge \\
 & \quad (\text{FORALL } (x\text{: Conf}): s(x) \Rightarrow \{\text{——}\}(F(\text{pl3}))(x)) \wedge \\
 & \quad\; (\text{FORALL } c\text{:} \\
 & \qquad s(c) \Rightarrow \\
 & \qquad\; (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c)) \text{ —- } ([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c))))
\end{array}
$$

Applying bddsimp,
we get 2 subgoals:

`partPlusTotalStrongerImpliesPart.1:`

$$
\begin{array}{|ll}
\{-1\} & \text{FORALL } (x\colon \text{Conf})\colon\ s(x) \Rightarrow \{\text{——}\}(F(\text{pl1}))(x) \\
\{-2\} & \text{FORALL } (x\colon \text{Conf})\colon\ s(x) \Rightarrow \{\text{——}\}(F(\text{pl2}))(x) \\
\{-3\} & \text{FORALL } c\colon \\
& \quad s(c) \Rightarrow \\
& \quad\ ((\text{[——]}(K(\text{pl1}))(A(\text{pl1}))(c)) \text{ —- } (\text{[——]}(K(\text{pl2}))(A(\text{pl2}))(c))) \\
\{-4\} & \text{FORALL } (c_1\colon \text{Conf})\colon \\
& \quad \{\text{——}\}(F(\text{pl2}))(c_1) \Rightarrow \\
& \quad\ (\{\text{——}\}(F(\text{pl3}))(c_1) \wedge \\
& \qquad ((\text{[——]}(K(\text{pl2}))(A(\text{pl2}))(c_1)) \text{ —- } (\text{[——]}(K(\text{pl3}))(A(\text{pl3}))(c_1)))) \\
\hline
\{1\} & \text{FORALL } c\colon \\
& \quad s(c) \Rightarrow \\
& \quad\ ((\text{[——]}(K(\text{pl1}))(A(\text{pl1}))(c)) \text{ —- } (\text{[——]}(K(\text{pl3}))(A(\text{pl3}))(c)))
\end{array}
$$

For the top quantifier in 1, we introduce Skolem constants: c,
`partPlusTotalStrongerImpliesPart.1:`

$$
\begin{array}{|ll}
\{-1\} & \text{FORALL } (x\colon \text{Conf})\colon\ s(x) \Rightarrow \{\text{——}\}(F(\text{pl1}))(x) \\
\{-2\} & \text{FORALL } (x\colon \text{Conf})\colon\ s(x) \Rightarrow \{\text{——}\}(F(\text{pl2}))(x) \\
\{-3\} & \text{FORALL } c\colon \\
& \quad s(c) \Rightarrow \\
& \quad\ ((\text{[——]}(K(\text{pl1}))(A(\text{pl1}))(c)) \text{ —- } (\text{[——]}(K(\text{pl2}))(A(\text{pl2}))(c))) \\
\{-4\} & \text{FORALL } (c_1\colon \text{Conf})\colon \\
& \quad \{\text{——}\}(F(\text{pl2}))(c_1) \Rightarrow \\
& \quad\ (\{\text{——}\}(F(\text{pl3}))(c_1) \wedge \\
& \qquad ((\text{[——]}(K(\text{pl2}))(A(\text{pl2}))(c_1)) \text{ —- } (\text{[——]}(K(\text{pl3}))(A(\text{pl3}))(c_1)))) \\
\hline
\{1\} & s(c) \Rightarrow ((\text{[——]}(K(\text{pl1}))(A(\text{pl1}))(c)) \text{ —- } (\text{[——]}(K(\text{pl3}))(A(\text{pl3}))(c)))
\end{array}
$$

Instantiating the top quantifier in -1 with the terms: c,
`partPlusTotalStrongerImpliesPart.1:`

$$
\begin{array}{|ll}
\{-1\} & s(c) \Rightarrow \{\text{——}\}(F(\text{pl1}))(c) \\
\{-2\} & \text{FORALL } (x\colon \text{Conf})\colon\ s(x) \Rightarrow \{\text{——}\}(F(\text{pl2}))(x) \\
\{-3\} & \text{FORALL } c\colon \\
& \quad s(c) \Rightarrow \\
& \quad\ ((\text{[——]}(K(\text{pl1}))(A(\text{pl1}))(c)) \text{ —- } (\text{[——]}(K(\text{pl2}))(A(\text{pl2}))(c))) \\
\{-4\} & \text{FORALL } (c_1\colon \text{Conf})\colon \\
& \quad \{\text{——}\}(F(\text{pl2}))(c_1) \Rightarrow \\
& \quad\ (\{\text{——}\}(F(\text{pl3}))(c_1) \wedge \\
& \qquad ((\text{[——]}(K(\text{pl2}))(A(\text{pl2}))(c_1)) \text{ —- } (\text{[——]}(K(\text{pl3}))(A(\text{pl3}))(c_1)))) \\
\hline
\{1\} & s(c) \Rightarrow ((\text{[——]}(K(\text{pl1}))(A(\text{pl1}))(c)) \text{ —- } (\text{[——]}(K(\text{pl3}))(A(\text{pl3}))(c)))
\end{array}
$$

Instantiating the top quantifier in -4 with the terms: c,

`partPlusTotalStrongerImpliesPart.1:`

$$
\begin{array}{l|ll}
& \{-1\} & s(c) \Rightarrow \{\text{——}\}(F(\mathrm{pl1}))(c) \\
& \{-2\} & \text{FORALL } (x\colon \mathrm{Conf})\colon s(x) \Rightarrow \{\text{——}\}(F(\mathrm{pl2}))(x) \\
& \{-3\} & \text{FORALL } c\colon \\
& & \quad s(c) \Rightarrow \\
& & \quad\; (([\text{——}](K(\mathrm{pl1}))(A(\mathrm{pl1}))(c)) \text{ —- } ([\text{——}](K(\mathrm{pl2}))(A(\mathrm{pl2}))(c))) \\
& \{-4\} & \{\text{——}\}(F(\mathrm{pl2}))(c) \Rightarrow \\
& & \; (\{\text{——}\}(F(\mathrm{pl3}))(c) \wedge \\
& & \quad\; (([\text{——}](K(\mathrm{pl2}))(A(\mathrm{pl2}))(c)) \text{ —- } ([\text{——}](K(\mathrm{pl3}))(A(\mathrm{pl3}))(c)))) \\
\hline
& \{1\} & s(c) \Rightarrow (([\text{——}](K(\mathrm{pl1}))(A(\mathrm{pl1}))(c)) \text{ —- } ([\text{——}](K(\mathrm{pl3}))(A(\mathrm{pl3}))(c)))
\end{array}
$$

Instantiating the top quantifier in -2 with the terms: c,
`partPlusTotalStrongerImpliesPart.1:`

$$
\begin{array}{l|ll}
& \{-1\} & s(c) \Rightarrow \{\text{——}\}(F(\mathrm{pl1}))(c) \\
& \{-2\} & s(c) \Rightarrow \{\text{——}\}(F(\mathrm{pl2}))(c) \\
& \{-3\} & \text{FORALL } c\colon \\
& & \quad s(c) \Rightarrow \\
& & \quad\; (([\text{——}](K(\mathrm{pl1}))(A(\mathrm{pl1}))(c)) \text{ —- } ([\text{——}](K(\mathrm{pl2}))(A(\mathrm{pl2}))(c))) \\
& \{-4\} & \{\text{——}\}(F(\mathrm{pl2}))(c) \Rightarrow \\
& & \; (\{\text{——}\}(F(\mathrm{pl3}))(c) \wedge \\
& & \quad\; (([\text{——}](K(\mathrm{pl2}))(A(\mathrm{pl2}))(c)) \text{ —- } ([\text{——}](K(\mathrm{pl3}))(A(\mathrm{pl3}))(c)))) \\
\hline
& \{1\} & s(c) \Rightarrow (([\text{——}](K(\mathrm{pl1}))(A(\mathrm{pl1}))(c)) \text{ —- } ([\text{——}](K(\mathrm{pl3}))(A(\mathrm{pl3}))(c)))
\end{array}
$$

Using lemma assetRefinement,
`partPlusTotalStrongerImpliesPart.1:`

$$
\begin{array}{l|ll}
& \{-1\} & \mathrm{orders}\big[\mathrm{set}\big[\mathrm{Asset}\big]\big].\mathrm{preorder?}(\text{—-}) \\
& \{-2\} & s(c) \Rightarrow \{\text{——}\}(F(\mathrm{pl1}))(c) \\
& \{-3\} & s(c) \Rightarrow \{\text{——}\}(F(\mathrm{pl2}))(c) \\
& \{-4\} & \text{FORALL } c\colon \\
& & \quad s(c) \Rightarrow \\
& & \quad\; (([\text{——}](K(\mathrm{pl1}))(A(\mathrm{pl1}))(c)) \text{ —- } ([\text{——}](K(\mathrm{pl2}))(A(\mathrm{pl2}))(c))) \\
& \{-5\} & \{\text{——}\}(F(\mathrm{pl2}))(c) \Rightarrow \\
& & \; (\{\text{——}\}(F(\mathrm{pl3}))(c) \wedge \\
& & \quad\; (([\text{——}](K(\mathrm{pl2}))(A(\mathrm{pl2}))(c)) \text{ —- } ([\text{——}](K(\mathrm{pl3}))(A(\mathrm{pl3}))(c)))) \\
\hline
& \{1\} & s(c) \Rightarrow (([\text{——}](K(\mathrm{pl1}))(A(\mathrm{pl1}))(c)) \text{ —- } ([\text{——}](K(\mathrm{pl3}))(A(\mathrm{pl3}))(c)))
\end{array}
$$

Expanding the definition of preorder?,

partPlusTotalStrongerImpliesPart.1:

{-1} reflexive?(—-) & transitive?(—-)
{-2} $s(c) \Rightarrow$ {——}$(F(\text{pl1}))(c)$
{-3} $s(c) \Rightarrow$ {——}$(F(\text{pl2}))(c)$
{-4} FORALL c:
  $s(c) \Rightarrow$
    (([——]$(K(\text{pl1}))(A(\text{pl1}))(c)$) —- ([——]$(K(\text{pl2}))(A(\text{pl2}))(c)$))
{-5} {——}$(F(\text{pl2}))(c) \Rightarrow$
  ({——}$(F(\text{pl3}))(c) \wedge$
    (([——]$(K(\text{pl2}))(A(\text{pl2}))(c)$) —- ([——]$(K(\text{pl3}))(A(\text{pl3}))(c)$)))

{1} $s(c) \Rightarrow$ (([——]$(K(\text{pl1}))(A(\text{pl1}))(c)$) —- ([——]$(K(\text{pl3}))(A(\text{pl3}))(c)$))

Applying disjunctive simplification to flatten sequent,
partPlusTotalStrongerImpliesPart.1:

{-1} reflexive?(—-)
{-2} transitive?(—-)
{-3} $s(c) \Rightarrow$ {——}$(F(\text{pl1}))(c)$
{-4} $s(c) \Rightarrow$ {——}$(F(\text{pl2}))(c)$
{-5} FORALL c:
  $s(c) \Rightarrow$
    (([——]$(K(\text{pl1}))(A(\text{pl1}))(c)$) —- ([——]$(K(\text{pl2}))(A(\text{pl2}))(c)$))
{-6} {——}$(F(\text{pl2}))(c) \Rightarrow$
  ({——}$(F(\text{pl3}))(c) \wedge$
    (([——]$(K(\text{pl2}))(A(\text{pl2}))(c)$) —- ([——]$(K(\text{pl3}))(A(\text{pl3}))(c)$)))
{-7} $s(c)$

{1} (([——]$(K(\text{pl1}))(A(\text{pl1}))(c)$) —- ([——]$(K(\text{pl3}))(A(\text{pl3}))(c)$))

Expanding the definition of transitive?,
partPlusTotalStrongerImpliesPart.1:

{-1} reflexive?(—-)
{-2} FORALL (x: set[Asset]), (y: set[Asset]), (z: set[Asset]):
  (x —- y) & (y —- z) $\Rightarrow$ (x —- z)
{-3} $s(c) \Rightarrow$ {——}$(F(\text{pl1}))(c)$
{-4} $s(c) \Rightarrow$ {——}$(F(\text{pl2}))(c)$
{-5} FORALL c:
  $s(c) \Rightarrow$
    (([——]$(K(\text{pl1}))(A(\text{pl1}))(c)$) —- ([——]$(K(\text{pl2}))(A(\text{pl2}))(c)$))
{-6} {——}$(F(\text{pl2}))(c) \Rightarrow$
  ({——}$(F(\text{pl3}))(c) \wedge$
    (([——]$(K(\text{pl2}))(A(\text{pl2}))(c)$) —- ([——]$(K(\text{pl3}))(A(\text{pl3}))(c)$)))
{-7} $s(c)$

{1} (([——]$(K(\text{pl1}))(A(\text{pl1}))(c)$) —- ([——]$(K(\text{pl3}))(A(\text{pl3}))(c)$))

Instantiating the top quantifier in -2 with the terms: $([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c))$, $([\text{——}](K(\text{pl2}))(A($
$([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c))$,

`partPlusTotalStrongerImpliesPart.1:`

$$
\begin{array}{ll}
\{\text{-1}\} & \text{reflexive?}(\text{—-}) \\
\{\text{-2}\} & (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c)) \text{ —- } ([\text{——}](K(\text{pl2}))(A(\text{pl2}))(c))) \ \& \\
 & \quad (([\text{——}](K(\text{pl2}))(A(\text{pl2}))(c)) \text{ —- } ([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c))) \\
 & \quad \Rightarrow (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c)) \text{ —- } ([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c))) \\
\{\text{-3}\} & s(c) \Rightarrow \{\text{——}\}(F(\text{pl1}))(c) \\
\{\text{-4}\} & s(c) \Rightarrow \{\text{——}\}(F(\text{pl2}))(c) \\
\{\text{-5}\} & \text{FORALL } c: \\
 & \quad s(c) \Rightarrow \\
 & \quad\quad (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c)) \text{ —- } ([\text{——}](K(\text{pl2}))(A(\text{pl2}))(c))) \\
\{\text{-6}\} & \{\text{——}\}(F(\text{pl2}))(c) \Rightarrow \\
 & \quad (\{\text{——}\}(F(\text{pl3}))(c) \wedge \\
 & \quad\quad (([\text{——}](K(\text{pl2}))(A(\text{pl2}))(c)) \text{ —- } ([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c)))) \\
\{\text{-7}\} & s(c) \\
\hline
\{1\} & (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c)) \text{ —- } ([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c)))
\end{array}
$$

Applying bddsimp,

`partPlusTotalStrongerImpliesPart.1:`

$$
\begin{array}{ll}
\{\text{-1}\} & \text{reflexive?}(\text{—-}) \\
\{\text{-2}\} & (([\text{——}](K(\text{pl2}))(A(\text{pl2}))(c)) \text{ —- } ([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c))) \\
\{\text{-3}\} & s(c) \\
\{\text{-4}\} & \{\text{——}\}(F(\text{pl1}))(c) \\
\{\text{-5}\} & \{\text{——}\}(F(\text{pl2}))(c) \\
\{\text{-6}\} & \text{FORALL } c: \\
 & \quad s(c) \Rightarrow \\
 & \quad\quad (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c)) \text{ —- } ([\text{——}](K(\text{pl2}))(A(\text{pl2}))(c))) \\
\{\text{-7}\} & \{\text{——}\}(F(\text{pl3}))(c) \\
\hline
\{1\} & (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c)) \text{ —- } ([\text{——}](K(\text{pl2}))(A(\text{pl2}))(c))) \\
\{2\} & (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c)) \text{ —- } ([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c)))
\end{array}
$$

Instantiating the top quantifier in -6 with the terms: c,

`partPlusTotalStrongerImpliesPart.1:`

$$
\begin{array}{ll}
\{\text{-1}\} & \text{reflexive?}(\text{—-}) \\
\{\text{-2}\} & (([\text{——}](K(\text{pl2}))(A(\text{pl2}))(c)) \text{ —- } ([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c))) \\
\{\text{-3}\} & s(c) \\
\{\text{-4}\} & \{\text{——}\}(F(\text{pl1}))(c) \\
\{\text{-5}\} & \{\text{——}\}(F(\text{pl2}))(c) \\
\{\text{-6}\} & s(c) \Rightarrow (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c)) \text{ —- } ([\text{——}](K(\text{pl2}))(A(\text{pl2}))(c))) \\
\{\text{-7}\} & \{\text{——}\}(F(\text{pl3}))(c) \\
\hline
\{1\} & (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c)) \text{ —- } ([\text{——}](K(\text{pl2}))(A(\text{pl2}))(c))) \\
\{2\} & (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c)) \text{ —- } ([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c)))
\end{array}
$$

Applying bddsimp,

This completes the proof of `partPlusTotalStrongerImpliesPart.1`.

`partPlusTotalStrongerImpliesPart.2`:

$$
\begin{array}{l|ll}
 & \{-1\} & \text{FORALL } (x\text{: Conf}):\ s(x) \Rightarrow \{\text{——}\}(F(\text{pl1}))(x) \\
 & \{-2\} & \text{FORALL } (x\text{: Conf}):\ s(x) \Rightarrow \{\text{——}\}(F(\text{pl2}))(x) \\
 & \{-3\} & \text{FORALL } c: \\
 & & \quad s(c) \Rightarrow \\
 & & \quad\ ((\text{[——]}(K(\text{pl1}))(A(\text{pl1}))(c)) \text{ —- } (\text{[——]}(K(\text{pl2}))(A(\text{pl2}))(c))) \\
 & \{-4\} & \text{FORALL } (c_1\text{: Conf}): \\
 & & \quad \{\text{——}\}(F(\text{pl2}))(c_1) \Rightarrow \\
 & & \quad\ (\{\text{——}\}(F(\text{pl3}))(c_1) \wedge \\
 & & \qquad (((\text{[——]}(K(\text{pl2}))(A(\text{pl2}))(c_1)) \text{ —- } (\text{[——]}(K(\text{pl3}))(A(\text{pl3}))(c_1)))) \\
\hline
 & \{1\} & \text{FORALL } (x\text{: Conf}):\ s(x) \Rightarrow \{\text{——}\}(F(\text{pl3}))(x)
\end{array}
$$

For the top quantifier in 1, we introduce Skolem constants: c,

`partPlusTotalStrongerImpliesPart.2`:

$$
\begin{array}{l|ll}
 & \{-1\} & \text{FORALL } (x\text{: Conf}):\ s(x) \Rightarrow \{\text{——}\}(F(\text{pl1}))(x) \\
 & \{-2\} & \text{FORALL } (x\text{: Conf}):\ s(x) \Rightarrow \{\text{——}\}(F(\text{pl2}))(x) \\
 & \{-3\} & \text{FORALL } c: \\
 & & \quad s(c) \Rightarrow \\
 & & \quad\ ((\text{[——]}(K(\text{pl1}))(A(\text{pl1}))(c)) \text{ —- } (\text{[——]}(K(\text{pl2}))(A(\text{pl2}))(c))) \\
 & \{-4\} & \text{FORALL } (c_1\text{: Conf}): \\
 & & \quad \{\text{——}\}(F(\text{pl2}))(c_1) \Rightarrow \\
 & & \quad\ (\{\text{——}\}(F(\text{pl3}))(c_1) \wedge \\
 & & \qquad (((\text{[——]}(K(\text{pl2}))(A(\text{pl2}))(c_1)) \text{ —- } (\text{[——]}(K(\text{pl3}))(A(\text{pl3}))(c_1)))) \\
\hline
 & \{1\} & s(c) \Rightarrow \{\text{——}\}(F(\text{pl3}))(c)
\end{array}
$$

Instantiating the top quantifier in -4 with the terms: c,

`partPlusTotalStrongerImpliesPart.2`:

$$
\begin{array}{l|ll}
 & \{-1\} & \text{FORALL } (x\text{: Conf}):\ s(x) \Rightarrow \{\text{——}\}(F(\text{pl1}))(x) \\
 & \{-2\} & \text{FORALL } (x\text{: Conf}):\ s(x) \Rightarrow \{\text{——}\}(F(\text{pl2}))(x) \\
 & \{-3\} & \text{FORALL } c: \\
 & & \quad s(c) \Rightarrow \\
 & & \quad\ ((\text{[——]}(K(\text{pl1}))(A(\text{pl1}))(c)) \text{ —- } (\text{[——]}(K(\text{pl2}))(A(\text{pl2}))(c))) \\
 & \{-4\} & \{\text{——}\}(F(\text{pl2}))(c) \Rightarrow \\
 & & \ (\{\text{——}\}(F(\text{pl3}))(c) \wedge \\
 & & \quad (((\text{[——]}(K(\text{pl2}))(A(\text{pl2}))(c)) \text{ —- } (\text{[——]}(K(\text{pl3}))(A(\text{pl3}))(c)))) \\
\hline
 & \{1\} & s(c) \Rightarrow \{\text{——}\}(F(\text{pl3}))(c)
\end{array}
$$

Instantiating the top quantifier in -2 with the terms: c,

`partPlusTotalStrongerImpliesPart.2`:

{-1} FORALL $(x\colon \text{Conf})\colon\ s(x) \Rightarrow \{\text{——}\}(F(\text{pl1}))(x)$
{-2} $s(c) \Rightarrow \{\text{——}\}(F(\text{pl2}))(c)$
{-3} FORALL c:
$s(c) \Rightarrow$
$(([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c))\ \text{—-}\ ([\text{——}](K(\text{pl2}))(A(\text{pl2}))(c)))$
{-4} $\{\text{——}\}(F(\text{pl2}))(c) \Rightarrow$
$(\{\text{——}\}(F(\text{pl3}))(c) \wedge$
$(([\text{——}](K(\text{pl2}))(A(\text{pl2}))(c))\ \text{—-}\ ([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c))))$

{1} $s(c) \Rightarrow \{\text{——}\}(F(\text{pl3}))(c)$

Instantiating the top quantifier in -3 with the terms: c,
`partPlusTotalStrongerImpliesPart.2`:

{-1} FORALL $(x\colon \text{Conf})\colon\ s(x) \Rightarrow \{\text{——}\}(F(\text{pl1}))(x)$
{-2} $s(c) \Rightarrow \{\text{——}\}(F(\text{pl2}))(c)$
{-3} $s(c) \Rightarrow (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c))\ \text{—-}\ ([\text{——}](K(\text{pl2}))(A(\text{pl2}))(c)))$
{-4} $\{\text{——}\}(F(\text{pl2}))(c) \Rightarrow$
$(\{\text{——}\}(F(\text{pl3}))(c) \wedge$
$(([\text{——}](K(\text{pl2}))(A(\text{pl2}))(c))\ \text{—-}\ ([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c))))$

{1} $s(c) \Rightarrow \{\text{——}\}(F(\text{pl3}))(c)$

Instantiating the top quantifier in -1 with the terms: c,
`partPlusTotalStrongerImpliesPart.2`:

{-1} $s(c) \Rightarrow \{\text{——}\}(F(\text{pl1}))(c)$
{-2} $s(c) \Rightarrow \{\text{——}\}(F(\text{pl2}))(c)$
{-3} $s(c) \Rightarrow (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c))\ \text{—-}\ ([\text{——}](K(\text{pl2}))(A(\text{pl2}))(c)))$
{-4} $\{\text{——}\}(F(\text{pl2}))(c) \Rightarrow$
$(\{\text{——}\}(F(\text{pl3}))(c) \wedge$
$(([\text{——}](K(\text{pl2}))(A(\text{pl2}))(c))\ \text{—-}\ ([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c))))$

{1} $s(c) \Rightarrow \{\text{——}\}(F(\text{pl3}))(c)$

Applying bddsimp,
This completes the proof of `partPlusTotalStrongerImpliesPart.2`.
Q.E.D.

Verbose proof for `commutableDiagram`.

`commutableDiagram`:

{1}	FORALL pl1, pl3, pl4, (s: set[Conf] | ($s \subseteq \{——\}(F(\text{pl1}))$)): (strongerPLrefinement(pl1, pl3) $\wedge$ strongPartialRefinement(pl3, pl4, s)) $\Rightarrow$ (EXISTS pl2: strongPartialRefinement(pl1, pl2, s) $\wedge$ strongerPLrefinement(pl2, pl4))

`commutableDiagram`:

{1}	FORALL pl1, pl3, pl4, (s: set[Conf] | ($s \subseteq \{——\}(F(\text{pl1}))$)): (strongerPLrefinement(pl1, pl3) $\wedge$ strongPartialRefinement(pl3, pl4, s)) $\Rightarrow$ (EXISTS pl2: strongPartialRefinement(pl1, pl2, s) $\wedge$ strongerPLrefinement(pl2, pl4))

For the top quantifier in 1, we introduce Skolem constants: (pl1 pl3 pl4 s),

`commutableDiagram`:

{1}	(strongerPLrefinement(pl1, pl3) $\wedge$ strongPartialRefinement(pl3, pl4, s)) $\Rightarrow$ (EXISTS pl2: strongPartialRefinement(pl1, pl2, s) $\wedge$ strongerPLrefinement(pl2, pl4))

Applying bddsimp,

`commutableDiagram`:

{-1}	strongerPLrefinement(pl1, pl3)
{-2}	strongPartialRefinement(pl3, pl4, s)
{1}	EXISTS pl2: strongPartialRefinement(pl1, pl2, s) $\wedge$ strongerPLrefinement(pl2, pl4)

Instantiating the top quantifier in 1 with the terms: pl4,

`commutableDiagram`:

{-1}	strongerPLrefinement(pl1, pl3)
{-2}	strongPartialRefinement(pl3, pl4, s)
{1}	strongPartialRefinement(pl1, pl4, s) $\wedge$ strongerPLrefinement(pl4, pl4)

Expanding the definition(s) of (strongerPLrefinement strongPartialRefinement), `commutableDiagram`:

```
{-1}  FORALL (c_1: Conf):
        {——}(F(pl1))(c_1) ⇒
         ({——}(F(pl3))(c_1) ∧
           (([——](K(pl1))(A(pl1))(c_1)) —- ([——](K(pl3))(A(pl3))(c_1))))
{-2}  (s ⊆ {——}(F(pl3))) ∧
       (s ⊆ {——}(F(pl4))) ∧ (FORALL c: s(c) ⇒ (prod(pl3, c) —-
      prod(pl4, c)))
-------------------------------------------------------------------------
{1}   ((s ⊆ {——}(F(pl4))) ∧
         (FORALL c: s(c) ⇒ (prod(pl1, c) —- prod(pl4, c))))
       ∧
       (FORALL (c_1: Conf):
            {——}(F(pl4))(c_1) ⇒
             ({——}(F(pl4))(c_1) ∧
               (([——](K(pl4))(A(pl4))(c_1)) —-
                 ([——](K(pl4))(A(pl4))(c_1)))))
```

Expanding the definition of subset?,
`commutableDiagram`:

```
{-1}  FORALL (c_1: Conf):
        {——}(F(pl1))(c_1) ⇒
         ({——}(F(pl3))(c_1) ∧
           (([——](K(pl1))(A(pl1))(c_1)) —- ([——](K(pl3))(A(pl3))(c_1))))
{-2}  (FORALL (x: Conf): (x ∈ s) ⇒ (x ∈ {——}(F(pl3)))) ∧
       (FORALL (x: Conf): (x ∈ s) ⇒ (x ∈ {——}(F(pl4)))) ∧
        (FORALL c: s(c) ⇒ (prod(pl3, c) —- prod(pl4, c)))
-------------------------------------------------------------------------
{1}   ((FORALL (x: Conf): (x ∈ s) ⇒ (x ∈ {——}(F(pl4)))) ∧
         (FORALL c: s(c) ⇒ (prod(pl1, c) —- prod(pl4, c))))
       ∧
       (FORALL (c_1: Conf):
            {——}(F(pl4))(c_1) ⇒
             ({——}(F(pl4))(c_1) ∧
               (([——](K(pl4))(A(pl4))(c_1)) —-
                 ([——](K(pl4))(A(pl4))(c_1)))))
```

Expanding the definition of member,

`commutableDiagram`:

$$
\begin{array}{l|ll}
 & \{-1\} & \text{FORALL } (c_1\text{: Conf}): \\
 & & \quad \{\text{——}\}(F(\text{pl1}))(c_1) \Rightarrow \\
 & & \quad\; (\{\text{——}\}(F(\text{pl3}))(c_1) \wedge \\
 & & \qquad ((([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c_1)) \text{ —- } ([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c_1)))) \\
 & \{-2\} & (\text{FORALL } (x\text{: Conf}): s(x) \Rightarrow \{\text{——}\}(F(\text{pl3}))(x)) \wedge \\
 & & \; (\text{FORALL } (x\text{: Conf}): s(x) \Rightarrow \{\text{——}\}(F(\text{pl4}))(x)) \wedge \\
 & & \quad (\text{FORALL } c\text{: } s(c) \Rightarrow (\text{prod(pl3, } c) \text{ —- } \text{prod(pl4, } c))) \\
\hline
 & \{1\} & ((\text{FORALL } (x\text{: Conf}): s(x) \Rightarrow \{\text{——}\}(F(\text{pl4}))(x)) \wedge \\
 & & \quad (\text{FORALL } c\text{: } s(c) \Rightarrow (\text{prod(pl1, } c) \text{ —- } \text{prod(pl4, } c)))) \\
 & & \; \wedge \\
 & & \; (\text{FORALL } (c_1\text{: Conf}): \\
 & & \qquad \{\text{——}\}(F(\text{pl4}))(c_1) \Rightarrow \\
 & & \qquad\; (\{\text{——}\}(F(\text{pl4}))(c_1) \wedge \\
 & & \qquad\quad ((([\text{——}](K(\text{pl4}))(A(\text{pl4}))(c_1)) \text{ —-} \\
 & & \qquad\qquad ([\text{——}](K(\text{pl4}))(A(\text{pl4}))(c_1)))))
\end{array}
$$

Applying bddsimp,
we get 2 subgoals:
`commutableDiagram.1`:

$$
\begin{array}{l|ll}
 & \{-1\} & \text{FORALL } (c_1\text{: Conf}): \\
 & & \quad \{\text{——}\}(F(\text{pl1}))(c_1) \Rightarrow \\
 & & \quad\; (\{\text{——}\}(F(\text{pl3}))(c_1) \wedge \\
 & & \qquad ((([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c_1)) \text{ —- } ([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c_1)))) \\
 & \{-2\} & \text{FORALL } (x\text{: Conf}): s(x) \Rightarrow \{\text{——}\}(F(\text{pl3}))(x) \\
 & \{-3\} & \text{FORALL } (x\text{: Conf}): s(x) \Rightarrow \{\text{——}\}(F(\text{pl4}))(x) \\
 & \{-4\} & \text{FORALL } c\text{: } s(c) \Rightarrow (\text{prod(pl3, } c) \text{ —- } \text{prod(pl4, } c)) \\
\hline
 & \{1\} & \text{FORALL } (c_1\text{: Conf}): \\
 & & \quad \{\text{——}\}(F(\text{pl4}))(c_1) \Rightarrow \\
 & & \quad\; (\{\text{——}\}(F(\text{pl4}))(c_1) \wedge \\
 & & \qquad ((([\text{——}](K(\text{pl4}))(A(\text{pl4}))(c_1)) \text{ —- } ([\text{——}](K(\text{pl4}))(A(\text{pl4}))(c_1))))
\end{array}
$$

For the top quantifier in 1, we introduce Skolem constants: c,

`commutableDiagram.1:`

{-1} FORALL (c_1: Conf):
{——}(F(pl1))(c_1) $\Rightarrow$
({——}(F(pl3))(c_1) $\wedge$
(([——](K(pl1))(A(pl1))(c_1)) —- ([——](K(pl3))(A(pl3))(c_1))))
{-2} FORALL (x: Conf): $s(x)$ $\Rightarrow$ {——}(F(pl3))(x)
{-3} FORALL (x: Conf): $s(x)$ $\Rightarrow$ {——}(F(pl4))(x)
{-4} FORALL c: $s(c)$ $\Rightarrow$ (prod(pl3, c) —- prod(pl4, c))

{1} {——}(F(pl4))(c) $\Rightarrow$
({——}(F(pl4))(c) $\wedge$
(([——](K(pl4))(A(pl4))(c)) —- ([——](K(pl4))(A(pl4))(c))))

Using lemma assetRefinement,
`commutableDiagram.1:`

{-1} orders[set[Asset]].preorder?(—-)
{-2} FORALL (c_1: Conf):
{——}(F(pl1))(c_1) $\Rightarrow$
({——}(F(pl3))(c_1) $\wedge$
(([——](K(pl1))(A(pl1))(c_1)) —- ([——](K(pl3))(A(pl3))(c_1))))
{-3} FORALL (x: Conf): $s(x)$ $\Rightarrow$ {——}(F(pl3))(x)
{-4} FORALL (x: Conf): $s(x)$ $\Rightarrow$ {——}(F(pl4))(x)
{-5} FORALL c: $s(c)$ $\Rightarrow$ (prod(pl3, c) —- prod(pl4, c))

{1} {——}(F(pl4))(c) $\Rightarrow$
({——}(F(pl4))(c) $\wedge$
(([——](K(pl4))(A(pl4))(c)) —- ([——](K(pl4))(A(pl4))(c))))

Expanding the definition of preorder?,
`commutableDiagram.1:`

{-1} reflexive?(—-) & transitive?(—-)
{-2} FORALL (c_1: Conf):
{——}(F(pl1))(c_1) $\Rightarrow$
({——}(F(pl3))(c_1) $\wedge$
(([——](K(pl1))(A(pl1))(c_1)) —- ([——](K(pl3))(A(pl3))(c_1))))
{-3} FORALL (x: Conf): $s(x)$ $\Rightarrow$ {——}(F(pl3))(x)
{-4} FORALL (x: Conf): $s(x)$ $\Rightarrow$ {——}(F(pl4))(x)
{-5} FORALL c: $s(c)$ $\Rightarrow$ (prod(pl3, c) —- prod(pl4, c))

{1} {——}(F(pl4))(c) $\Rightarrow$
({——}(F(pl4))(c) $\wedge$
(([——](K(pl4))(A(pl4))(c)) —- ([——](K(pl4))(A(pl4))(c))))

Applying disjunctive simplification to flatten sequent,

`commutableDiagram.1:`

```
{-1}  reflexive?(—-)
{-2}  transitive?(—-)
{-3}  FORALL (c_1: Conf):
        {——}(F(pl1))(c_1) ⇒
         ({——}(F(pl3))(c_1) ∧
           (([——](K(pl1))(A(pl1))(c_1)) —- ([——](K(pl3))(A(pl3))(c_1))))
{-4}  FORALL (x: Conf): s(x) ⇒ {——}(F(pl3))(x)
{-5}  FORALL (x: Conf): s(x) ⇒ {——}(F(pl4))(x)
{-6}  FORALL c: s(c) ⇒ (prod(pl3, c) —- prod(pl4, c))
{-7}  {——}(F(pl4))(c)
-------------------------------------------------------------
{1}   ({——}(F(pl4))(c) ∧
         (([——](K(pl4))(A(pl4))(c)) —- ([——](K(pl4))(A(pl4))(c))))
```

Expanding the definition of reflexive?,
`commutableDiagram.1:`

```
{-1}  FORALL (x: set[Asset]): (x —- x)
{-2}  transitive?(—-)
{-3}  FORALL (c_1: Conf):
        {——}(F(pl1))(c_1) ⇒
         ({——}(F(pl3))(c_1) ∧
           (([——](K(pl1))(A(pl1))(c_1)) —- ([——](K(pl3))(A(pl3))(c_1))))
{-4}  FORALL (x: Conf): s(x) ⇒ {——}(F(pl3))(x)
{-5}  FORALL (x: Conf): s(x) ⇒ {——}(F(pl4))(x)
{-6}  FORALL c: s(c) ⇒ (prod(pl3, c) —- prod(pl4, c))
{-7}  {——}(F(pl4))(c)
-------------------------------------------------------------
{1}   ({——}(F(pl4))(c) ∧
         (([——](K(pl4))(A(pl4))(c)) —- ([——](K(pl4))(A(pl4))(c))))
```

Instantiating the top quantifier in -1 with the terms: ([——](K(pl4))(A(pl4))(c)),
`commutableDiagram.1:`

```
{-1}  (([——](K(pl4))(A(pl4))(c)) —- ([——](K(pl4))(A(pl4))(c)))
{-2}  transitive?(—-)
{-3}  FORALL (c_1: Conf):
        {——}(F(pl1))(c_1) ⇒
         ({——}(F(pl3))(c_1) ∧
           (([——](K(pl1))(A(pl1))(c_1)) —- ([——](K(pl3))(A(pl3))(c_1))))
{-4}  FORALL (x: Conf): s(x) ⇒ {——}(F(pl3))(x)
{-5}  FORALL (x: Conf): s(x) ⇒ {——}(F(pl4))(x)
{-6}  FORALL c: s(c) ⇒ (prod(pl3, c) —- prod(pl4, c))
{-7}  {——}(F(pl4))(c)
-------------------------------------------------------------
{1}   ({——}(F(pl4))(c) ∧
         (([——](K(pl4))(A(pl4))(c)) —- ([——](K(pl4))(A(pl4))(c))))
```

Simplifying, rewriting, and recording with decision procedures,
This completes the proof of `commutableDiagram.1`.
`commutableDiagram.2`:

$$
\begin{array}{|ll}
\{\text{-1}\} & \text{FORALL } (c_1\text{: Conf}):\\
& \quad \{\text{——}\}(F(\text{pl1}))(c_1) \Rightarrow\\
& \quad\; (\{\text{——}\}(F(\text{pl3}))(c_1) \wedge\\
& \qquad\; (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c_1)) \text{ —- } ([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c_1))))\\
\{\text{-2}\} & \text{FORALL } (x\text{: Conf}):\ s(x) \Rightarrow \{\text{——}\}(F(\text{pl3}))(x)\\
\{\text{-3}\} & \text{FORALL } (x\text{: Conf}):\ s(x) \Rightarrow \{\text{——}\}(F(\text{pl4}))(x)\\
\{\text{-4}\} & \text{FORALL } c\text{: } s(c) \Rightarrow (\text{prod(pl3, } c) \text{ —- } \text{prod(pl4, } c))\\
\hline
\{1\} & \text{FORALL } c\text{: } s(c) \Rightarrow (\text{prod(pl1, } c) \text{ —- } \text{prod(pl4, } c))
\end{array}
$$

For the top quantifier in 1, we introduce Skolem constants: c,
`commutableDiagram.2`:

$$
\begin{array}{|ll}
\{\text{-1}\} & \text{FORALL } (c_1\text{: Conf}):\\
& \quad \{\text{——}\}(F(\text{pl1}))(c_1) \Rightarrow\\
& \quad\; (\{\text{——}\}(F(\text{pl3}))(c_1) \wedge\\
& \qquad\; (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c_1)) \text{ —- } ([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c_1))))\\
\{\text{-2}\} & \text{FORALL } (x\text{: Conf}):\ s(x) \Rightarrow \{\text{——}\}(F(\text{pl3}))(x)\\
\{\text{-3}\} & \text{FORALL } (x\text{: Conf}):\ s(x) \Rightarrow \{\text{——}\}(F(\text{pl4}))(x)\\
\{\text{-4}\} & \text{FORALL } c\text{: } s(c) \Rightarrow (\text{prod(pl3, } c) \text{ —- } \text{prod(pl4, } c))\\
\hline
\{1\} & s(c) \Rightarrow (\text{prod(pl1, } c) \text{ —- } \text{prod(pl4, } c))
\end{array}
$$

Expanding the definition of prod,
`commutableDiagram.2`:

$$
\begin{array}{|ll}
\{\text{-1}\} & \text{FORALL } (c_1\text{: Conf}):\\
& \quad \{\text{——}\}(F(\text{pl1}))(c_1) \Rightarrow\\
& \quad\; (\{\text{——}\}(F(\text{pl3}))(c_1) \wedge\\
& \qquad\; (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c_1)) \text{ —- } ([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c_1))))\\
\{\text{-2}\} & \text{FORALL } (x\text{: Conf}):\ s(x) \Rightarrow \{\text{——}\}(F(\text{pl3}))(x)\\
\{\text{-3}\} & \text{FORALL } (x\text{: Conf}):\ s(x) \Rightarrow \{\text{——}\}(F(\text{pl4}))(x)\\
\{\text{-4}\} & \text{FORALL } c\text{:}\\
& \quad s(c) \Rightarrow\\
& \quad\; (([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c)) \text{ —- } ([\text{——}](K(\text{pl4}))(A(\text{pl4}))(c)))\\
\hline
\{1\} & s(c) \Rightarrow (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c)) \text{ —- } ([\text{——}](K(\text{pl4}))(A(\text{pl4}))(c)))
\end{array}
$$

Instantiating the top quantifier in -1 with the terms: c,

`commutableDiagram.2:`

$$
\begin{array}{|ll}
\{\text{-}1\} & \{\text{——}\}(F(\mathrm{pl1}))(c) \Rightarrow \\
& \quad (\{\text{——}\}(F(\mathrm{pl3}))(c) \wedge \\
& \qquad (([\text{——}](K(\mathrm{pl1}))(A(\mathrm{pl1}))(c)) \text{ —- } ([\text{——}](K(\mathrm{pl3}))(A(\mathrm{pl3}))(c)))) \\
\{\text{-}2\} & \text{FORALL } (x\text{: Conf}):\ s(x) \Rightarrow \{\text{——}\}(F(\mathrm{pl3}))(x) \\
\{\text{-}3\} & \text{FORALL } (x\text{: Conf}):\ s(x) \Rightarrow \{\text{——}\}(F(\mathrm{pl4}))(x) \\
\{\text{-}4\} & \text{FORALL } c: \\
& \quad s(c) \Rightarrow \\
& \quad (([\text{——}](K(\mathrm{pl3}))(A(\mathrm{pl3}))(c)) \text{ —- } ([\text{——}](K(\mathrm{pl4}))(A(\mathrm{pl4}))(c))) \\
\hline
\{1\} & s(c) \Rightarrow (([\text{——}](K(\mathrm{pl1}))(A(\mathrm{pl1}))(c)) \text{ —- } ([\text{——}](K(\mathrm{pl4}))(A(\mathrm{pl4}))(c)))
\end{array}
$$

Instantiating the top quantifier in -2 with the terms: c,
`commutableDiagram.2:`

$$
\begin{array}{|ll}
\{\text{-}1\} & \{\text{——}\}(F(\mathrm{pl1}))(c) \Rightarrow \\
& \quad (\{\text{——}\}(F(\mathrm{pl3}))(c) \wedge \\
& \qquad (([\text{——}](K(\mathrm{pl1}))(A(\mathrm{pl1}))(c)) \text{ —- } ([\text{——}](K(\mathrm{pl3}))(A(\mathrm{pl3}))(c)))) \\
\{\text{-}2\} & s(c) \Rightarrow \{\text{——}\}(F(\mathrm{pl3}))(c) \\
\{\text{-}3\} & \text{FORALL } (x\text{: Conf}):\ s(x) \Rightarrow \{\text{——}\}(F(\mathrm{pl4}))(x) \\
\{\text{-}4\} & \text{FORALL } c: \\
& \quad s(c) \Rightarrow \\
& \quad (([\text{——}](K(\mathrm{pl3}))(A(\mathrm{pl3}))(c)) \text{ —- } ([\text{——}](K(\mathrm{pl4}))(A(\mathrm{pl4}))(c))) \\
\hline
\{1\} & s(c) \Rightarrow (([\text{——}](K(\mathrm{pl1}))(A(\mathrm{pl1}))(c)) \text{ —- } ([\text{——}](K(\mathrm{pl4}))(A(\mathrm{pl4}))(c)))
\end{array}
$$

Instantiating the top quantifier in -4 with the terms: c,
`commutableDiagram.2:`

$$
\begin{array}{|ll}
\{\text{-}1\} & \{\text{——}\}(F(\mathrm{pl1}))(c) \Rightarrow \\
& \quad (\{\text{——}\}(F(\mathrm{pl3}))(c) \wedge \\
& \qquad (([\text{——}](K(\mathrm{pl1}))(A(\mathrm{pl1}))(c)) \text{ —- } ([\text{——}](K(\mathrm{pl3}))(A(\mathrm{pl3}))(c)))) \\
\{\text{-}2\} & s(c) \Rightarrow \{\text{——}\}(F(\mathrm{pl3}))(c) \\
\{\text{-}3\} & \text{FORALL } (x\text{: Conf}):\ s(x) \Rightarrow \{\text{——}\}(F(\mathrm{pl4}))(x) \\
\{\text{-}4\} & s(c) \Rightarrow (([\text{——}](K(\mathrm{pl3}))(A(\mathrm{pl3}))(c)) \text{ —- } ([\text{——}](K(\mathrm{pl4}))(A(\mathrm{pl4}))(c))) \\
\hline
\{1\} & s(c) \Rightarrow (([\text{——}](K(\mathrm{pl1}))(A(\mathrm{pl1}))(c)) \text{ —- } ([\text{——}](K(\mathrm{pl4}))(A(\mathrm{pl4}))(c)))
\end{array}
$$

Using lemma assetRefinement,
`commutableDiagram.2:`

$$
\begin{array}{|ll}
\{\text{-}1\} & \text{orders}\big[\text{set}\big[\text{Asset}\big]\big]\text{.preorder?(—-)} \\
\{\text{-}2\} & \{\text{——}\}(F(\mathrm{pl1}))(c) \Rightarrow \\
& \quad (\{\text{——}\}(F(\mathrm{pl3}))(c) \wedge \\
& \qquad (([\text{——}](K(\mathrm{pl1}))(A(\mathrm{pl1}))(c)) \text{ —- } ([\text{——}](K(\mathrm{pl3}))(A(\mathrm{pl3}))(c)))) \\
\{\text{-}3\} & s(c) \Rightarrow \{\text{——}\}(F(\mathrm{pl3}))(c) \\
\{\text{-}4\} & \text{FORALL } (x\text{: Conf}):\ s(x) \Rightarrow \{\text{——}\}(F(\mathrm{pl4}))(x) \\
\{\text{-}5\} & s(c) \Rightarrow (([\text{——}](K(\mathrm{pl3}))(A(\mathrm{pl3}))(c)) \text{ —- } ([\text{——}](K(\mathrm{pl4}))(A(\mathrm{pl4}))(c))) \\
\hline
\{1\} & s(c) \Rightarrow (([\text{——}](K(\mathrm{pl1}))(A(\mathrm{pl1}))(c)) \text{ —- } ([\text{——}](K(\mathrm{pl4}))(A(\mathrm{pl4}))(c)))
\end{array}
$$

Expanding the definition of preorder?,
`commutableDiagram.2:`

{-1}	reflexive?(—-) & transitive?(—-)
{-2}	{——}(F(pl1))(c) ⇒ ({——}(F(pl3))(c) ∧ (([——](K(pl1))(A(pl1))(c)) —- ([——](K(pl3))(A(pl3))(c))))
{-3}	$s(c)$ ⇒ {——}(F(pl3))(c)
{-4}	FORALL (x: Conf): $s(x)$ ⇒ {——}(F(pl4))(x)
{-5}	$s(c)$ ⇒ (([——](K(pl3))(A(pl3))(c)) —- ([——](K(pl4))(A(pl4))(c)))
{1}	$s(c)$ ⇒ (([——](K(pl1))(A(pl1))(c)) —- ([——](K(pl4))(A(pl4))(c)))

Applying disjunctive simplification to flatten sequent,
`commutableDiagram.2:`

{-1}	reflexive?(—-)
{-2}	transitive?(—-)
{-3}	{——}(F(pl1))(c) ⇒ ({——}(F(pl3))(c) ∧ (([——](K(pl1))(A(pl1))(c)) —- ([——](K(pl3))(A(pl3))(c))))
{-4}	$s(c)$ ⇒ {——}(F(pl3))(c)
{-5}	FORALL (x: Conf): $s(x)$ ⇒ {——}(F(pl4))(x)
{-6}	$s(c)$ ⇒ (([——](K(pl3))(A(pl3))(c)) —- ([——](K(pl4))(A(pl4))(c)))
{-7}	$s(c)$
{1}	(([——](K(pl1))(A(pl1))(c)) —- ([——](K(pl4))(A(pl4))(c)))

Expanding the definition of transitive?,
`commutableDiagram.2:`

{-1}	reflexive?(—-)
{-2}	FORALL (x: set[Asset]), (y: set[Asset]), (z: set[Asset]): (x —- y) & (y —- z) ⇒ (x —- z)
{-3}	{——}(F(pl1))(c) ⇒ ({——}(F(pl3))(c) ∧ (([——](K(pl1))(A(pl1))(c)) —- ([——](K(pl3))(A(pl3))(c))))
{-4}	$s(c)$ ⇒ {——}(F(pl3))(c)
{-5}	FORALL (x: Conf): $s(x)$ ⇒ {——}(F(pl4))(x)
{-6}	$s(c)$ ⇒ (([——](K(pl3))(A(pl3))(c)) —- ([——](K(pl4))(A(pl4))(c)))
{-7}	$s(c)$
{1}	(([——](K(pl1))(A(pl1))(c)) —- ([——](K(pl4))(A(pl4))(c)))

Instantiating the top quantifier in -2 with the terms: ([——](K(pl1))(A(pl1))(c)), ([——](K(pl3))(A(
([——](K(pl4))(A(pl4))(c)),

`commutableDiagram.2:`

{-1}	reflexive?(—-)
{-2}	$(([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c))$ —- $([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c)))$ & $(([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c))$ —- $([\text{——}](K(\text{pl4}))(A(\text{pl4}))(c)))$ $\Rightarrow (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c))$ —- $([\text{——}](K(\text{pl4}))(A(\text{pl4}))(c)))$
{-3}	$\{\text{——}\}(F(\text{pl1}))(c) \Rightarrow$ $(\{\text{——}\}(F(\text{pl3}))(c) \wedge$ $(([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c))$ —- $([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c))))$
{-4}	$s(c) \Rightarrow \{\text{——}\}(F(\text{pl3}))(c)$
{-5}	FORALL $(x\colon$ Conf$)\colon\ s(x) \Rightarrow \{\text{——}\}(F(\text{pl4}))(x)$
{-6}	$s(c) \Rightarrow (([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c))$ —- $([\text{——}](K(\text{pl4}))(A(\text{pl4}))(c)))$
{-7}	$s(c)$
{1}	$(([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c))$ —- $([\text{——}](K(\text{pl4}))(A(\text{pl4}))(c)))$

Applying bddsimp,
`commutableDiagram.2:`

{-1}	reflexive?(—-)
{-2}	$(([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c))$ —- $([\text{——}](K(\text{pl4}))(A(\text{pl4}))(c)))$
{-3}	$\{\text{——}\}(F(\text{pl3}))(c)$
{-4}	$s(c)$
{-5}	FORALL $(x\colon$ Conf$)\colon\ s(x) \Rightarrow \{\text{——}\}(F(\text{pl4}))(x)$
{1}	$(([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c))$ —- $([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c)))$
{2}	$(([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c))$ —- $([\text{——}](K(\text{pl4}))(A(\text{pl4}))(c)))$
{3}	$\{\text{——}\}(F(\text{pl1}))(c)$

Instantiating the top quantifier in -5 with the terms: c,
`commutableDiagram.2:`

{-1}	reflexive?(—-)
{-2}	$(([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c))$ —- $([\text{——}](K(\text{pl4}))(A(\text{pl4}))(c)))$
{-3}	$\{\text{——}\}(F(\text{pl3}))(c)$
{-4}	$s(c)$
{-5}	$s(c) \Rightarrow \{\text{——}\}(F(\text{pl4}))(c)$
{1}	$(([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c))$ —- $([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c)))$
{2}	$(([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c))$ —- $([\text{——}](K(\text{pl4}))(A(\text{pl4}))(c)))$
{3}	$\{\text{——}\}(F(\text{pl1}))(c)$

Adding type constraints for s,

`commutableDiagram.2:`

$$
\begin{array}{|ll}
\{\text{-1}\} & (s \subseteq \{\text{——}\}(F(\text{pl1}))) \\
\{\text{-2}\} & \text{reflexive?}(\text{—-}) \\
\{\text{-3}\} & (([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c)) \ \text{—-} \ ([\text{——}](K(\text{pl4}))(A(\text{pl4}))(c))) \\
\{\text{-4}\} & \{\text{——}\}(F(\text{pl3}))(c) \\
\{\text{-5}\} & s(c) \\
\{\text{-6}\} & s(c) \ \Rightarrow \ \{\text{——}\}(F(\text{pl4}))(c) \\
\hline
\{1\} & (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c)) \ \text{—-} \ ([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c))) \\
\{2\} & (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c)) \ \text{—-} \ ([\text{——}](K(\text{pl4}))(A(\text{pl4}))(c))) \\
\{3\} & \{\text{——}\}(F(\text{pl1}))(c)
\end{array}
$$

Expanding the definition of subset?,
`commutableDiagram.2:`

$$
\begin{array}{|ll}
\{\text{-1}\} & \text{FORALL } (x\text{: Conf})\text{: } (x \in s) \ \Rightarrow \ (x \in \{\text{——}\}(F(\text{pl1}))) \\
\{\text{-2}\} & \text{reflexive?}(\text{—-}) \\
\{\text{-3}\} & (([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c)) \ \text{—-} \ ([\text{——}](K(\text{pl4}))(A(\text{pl4}))(c))) \\
\{\text{-4}\} & \{\text{——}\}(F(\text{pl3}))(c) \\
\{\text{-5}\} & s(c) \\
\{\text{-6}\} & s(c) \ \Rightarrow \ \{\text{——}\}(F(\text{pl4}))(c) \\
\hline
\{1\} & (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c)) \ \text{—-} \ ([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c))) \\
\{2\} & (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c)) \ \text{—-} \ ([\text{——}](K(\text{pl4}))(A(\text{pl4}))(c))) \\
\{3\} & \{\text{——}\}(F(\text{pl1}))(c)
\end{array}
$$

Instantiating the top quantifier in -1 with the terms: c,
`commutableDiagram.2:`

$$
\begin{array}{|ll}
\{\text{-1}\} & (c \in s) \ \Rightarrow \ (c \in \{\text{——}\}(F(\text{pl1}))) \\
\{\text{-2}\} & \text{reflexive?}(\text{—-}) \\
\{\text{-3}\} & (([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c)) \ \text{—-} \ ([\text{——}](K(\text{pl4}))(A(\text{pl4}))(c))) \\
\{\text{-4}\} & \{\text{——}\}(F(\text{pl3}))(c) \\
\{\text{-5}\} & s(c) \\
\{\text{-6}\} & s(c) \ \Rightarrow \ \{\text{——}\}(F(\text{pl4}))(c) \\
\hline
\{1\} & (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c)) \ \text{—-} \ ([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c))) \\
\{2\} & (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c)) \ \text{—-} \ ([\text{——}](K(\text{pl4}))(A(\text{pl4}))(c))) \\
\{3\} & \{\text{——}\}(F(\text{pl1}))(c)
\end{array}
$$

Expanding the definition of member,

`commutableDiagram.2`:

$$
\begin{array}{|ll}
\{\text{-1}\} & s(c) \;\Rightarrow\; \{\text{——}\}(F(\text{pl1}))(c) \\
\{\text{-2}\} & \text{reflexive?}(\text{—-}) \\
\{\text{-3}\} & (([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c)) \;\text{—-}\; ([\text{——}](K(\text{pl4}))(A(\text{pl4}))(c))) \\
\{\text{-4}\} & \{\text{——}\}(F(\text{pl3}))(c) \\
\{\text{-5}\} & s(c) \\
\{\text{-6}\} & s(c) \;\Rightarrow\; \{\text{——}\}(F(\text{pl4}))(c) \\
\hline
\{1\} & (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c)) \;\text{—-}\; ([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c))) \\
\{2\} & (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c)) \;\text{—-}\; ([\text{——}](K(\text{pl4}))(A(\text{pl4}))(c))) \\
\{3\} & \{\text{——}\}(F(\text{pl1}))(c)
\end{array}
$$

Applying bddsimp,
This completes the proof of `commutableDiagram.2`.
Q.E.D.

Verbose proof for `commutableDiagram2`.
`commutableDiagram2`:

{1}	FORALL pl1, pl2, pl4, s: (strongPartialRefinement(pl1, pl2, s) $\wedge$ strongerPLrefinement(pl2, pl4)) $\Rightarrow$ (EXISTS pl3: strongerPLrefinement(pl1, pl3) $\wedge$ strongPartialRefinement(pl3, pl4, s))

`commutableDiagram2`:

{1}	FORALL pl1, pl2, pl4, s: (strongPartialRefinement(pl1, pl2, s) $\wedge$ strongerPLrefinement(pl2, pl4)) $\Rightarrow$ (EXISTS pl3: strongerPLrefinement(pl1, pl3) $\wedge$ strongPartialRefinement(pl3, pl4, s))

For the top quantifier in 1, we introduce Skolem constants: (pl1 pl2 pl4 s),
`commutableDiagram2`:

{1}	(strongPartialRefinement(pl1, pl2, s) $\wedge$ strongerPLrefinement(pl2, pl4)) $\Rightarrow$ (EXISTS pl3: strongerPLrefinement(pl1, pl3) $\wedge$ strongPartialRefinement(pl3, pl4, s))

Applying bddsimp,
`commutableDiagram2`:

{-1}	strongPartialRefinement(pl1, pl2, s)
{-2}	strongerPLrefinement(pl2, pl4)
{1}	EXISTS pl3: strongerPLrefinement(pl1, pl3) $\wedge$ strongPartialRefinement(pl3, pl4, s)

Instantiating the top quantifier in 1 with the terms: pl1,
`commutableDiagram2`:

{-1}	strongPartialRefinement(pl1, pl2, s)
{-2}	strongerPLrefinement(pl2, pl4)
{1}	strongerPLrefinement(pl1, pl1) $\wedge$ strongPartialRefinement(pl1, pl4, s)

Applying bddsimp,
we get 2 subgoals:

`commutableDiagram2.1:`

{-1}	strongPartialRefinement(pl1, pl2, s)
{-2}	strongerPLrefinement(pl2, pl4)
{1}	strongPartialRefinement(pl1, pl4, s)

Applying totalImpliesPartial
`commutableDiagram2.1:`

{-1}	$\forall$ (pl1, pl2, s: set[Conf] | ($s \subseteq$ {——}(F(pl1)))): strongerPLrefinement(pl1, pl2) $\Rightarrow$ strongPartialRefinement(pl1, pl2, s)
{-2}	strongPartialRefinement(pl1, pl2, s)
{-3}	strongerPLrefinement(pl2, pl4)
{1}	strongPartialRefinement(pl1, pl4, s)

Instantiating the top quantifier in -1 with the terms: pl2, pl4, s,
we get 2 subgoals:
`commutableDiagram2.1.1:`

{-1}	strongerPLrefinement(pl2, pl4) $\Rightarrow$ strongPartialRefinement(pl2, pl4, s)
{-2}	strongPartialRefinement(pl1, pl2, s)
{-3}	strongerPLrefinement(pl2, pl4)
{1}	strongPartialRefinement(pl1, pl4, s)

Applying bddsimp,
`commutableDiagram2.1.1:`

{-1}	strongerPLrefinement(pl2, pl4)
{-2}	strongPartialRefinement(pl2, pl4, s)
{-3}	strongPartialRefinement(pl1, pl2, s)
{1}	strongPartialRefinement(pl1, pl4, s)

Applying strongPartRefTransitive
`commutableDiagram2.1.1:`

{-1}	$\forall$ (pl1, pl2, pl3, s, t): (strongPartialRefinement(pl1, pl2, s) $\wedge$ strongPartialRefinement(pl2, pl3, t)) $\Rightarrow$ strongPartialRefinement(pl1, pl3, ($s \cap t$))
{-2}	strongerPLrefinement(pl2, pl4)
{-3}	strongPartialRefinement(pl2, pl4, s)
{-4}	strongPartialRefinement(pl1, pl2, s)
{1}	strongPartialRefinement(pl1, pl4, s)

Instantiating the top quantifier in -1 with the terms: pl1, pl2, pl4, s, s, `commutableDiagram2.1.1:`

{-1}	(strongPartialRefinement(pl1, pl2, s) ∧ strongPartialRefinement(pl2, pl4, s)) ⇒ strongPartialRefinement(pl1, pl4, $(s \cap s)$)
{-2}	strongerPLrefinement(pl2, pl4)
{-3}	strongPartialRefinement(pl2, pl4, s)
{-4}	strongPartialRefinement(pl1, pl2, s)
{1}	strongPartialRefinement(pl1, pl4, s)

Applying bddsimp,
`commutableDiagram2.1.1:`

{-1}	strongPartialRefinement(pl1, pl2, s)
{-2}	strongPartialRefinement(pl2, pl4, s)
{-3}	strongPartialRefinement(pl1, pl4, $(s \cap s)$)
{-4}	strongerPLrefinement(pl2, pl4)
{1}	strongPartialRefinement(pl1, pl4, s)

Applying sets_lemmas[Conf].intersection_idempotent
`commutableDiagram2.1.1:`

{-1}	∀ (a: set[Conf]): $(a \cap a) = a$
{-2}	strongPartialRefinement(pl1, pl2, s)
{-3}	strongPartialRefinement(pl2, pl4, s)
{-4}	strongPartialRefinement(pl1, pl4, $(s \cap s)$)
{-5}	strongerPLrefinement(pl2, pl4)
{1}	strongPartialRefinement(pl1, pl4, s)

Instantiating the top quantifier in -1 with the terms: s,
`commutableDiagram2.1.1:`

{-1}	$(s \cap s) = s$
{-2}	strongPartialRefinement(pl1, pl2, s)
{-3}	strongPartialRefinement(pl2, pl4, s)
{-4}	strongPartialRefinement(pl1, pl4, $(s \cap s)$)
{-5}	strongerPLrefinement(pl2, pl4)
{1}	strongPartialRefinement(pl1, pl4, s)

Replacing using formula -1,

`commutableDiagram2.1.1:`

$$\begin{array}{r|l}
\{-1\} & (s \cap s) = s \\
\{-2\} & \text{strongPartialRefinement}(\text{pl1}, \text{pl2}, s) \\
\{-3\} & \text{strongPartialRefinement}(\text{pl2}, \text{pl4}, s) \\
\{-4\} & \text{strongPartialRefinement}(\text{pl1}, \text{pl4}, s) \\
\{-5\} & \text{strongerPLrefinement}(\text{pl2}, \text{pl4}) \\
\hline
\{1\} & \text{strongPartialRefinement}(\text{pl1}, \text{pl4}, s)
\end{array}$$

which is trivially true.
This completes the proof of `commutableDiagram2.1.1`.
`commutableDiagram2.1.2:`

$$\begin{array}{r|l}
\{-1\} & \text{strongPartialRefinement}(\text{pl1}, \text{pl2}, s) \\
\{-2\} & \text{strongerPLrefinement}(\text{pl2}, \text{pl4}) \\
\hline
\{1\} & (s \subseteq \{\text{——}\}(F(\text{pl2}))) \\
\{2\} & \text{strongPartialRefinement}(\text{pl1}, \text{pl4}, s)
\end{array}$$

Expanding the definition of strongPartialRefinement,
`commutableDiagram2.1.2:`

$$\begin{array}{r|l}
\{-1\} & (s \subseteq \{\text{——}\}(F(\text{pl1}))) \wedge \\
 & \quad (s \subseteq \{\text{——}\}(F(\text{pl2}))) \wedge (\textsc{forall}\ c\text{:}\ s(c) \Rightarrow (\text{prod}(\text{pl1}, c) \text{ —-} \\
 & \text{prod}(\text{pl2}, c))) \\
\{-2\} & \text{strongerPLrefinement}(\text{pl2}, \text{pl4}) \\
\hline
\{1\} & (s \subseteq \{\text{——}\}(F(\text{pl2}))) \\
\{2\} & (s \subseteq \{\text{——}\}(F(\text{pl1}))) \wedge \\
 & \quad (s \subseteq \{\text{——}\}(F(\text{pl4}))) \wedge (\textsc{forall}\ c\text{:}\ s(c) \Rightarrow (\text{prod}(\text{pl1}, c) \text{ —-} \\
 & \text{prod}(\text{pl4}, c)))
\end{array}$$

Applying bddsimp,
This completes the proof of `commutableDiagram2.1.2`.
`commutableDiagram2.2:`

$$\begin{array}{r|l}
\{-1\} & \text{strongPartialRefinement}(\text{pl1}, \text{pl2}, s) \\
\{-2\} & \text{strongerPLrefinement}(\text{pl2}, \text{pl4}) \\
\hline
\{1\} & \text{strongerPLrefinement}(\text{pl1}, \text{pl1})
\end{array}$$

Applying strongerPLref
`commutableDiagram2.2:`

$$\begin{array}{r|l}
\{-1\} & \text{orders}\big[\text{PL}\big[\text{Conf}, \text{FM}, \text{Asset}, \text{AssetName}, \text{CK}, \{\text{——}\}, [\text{——}]\big]\big]\text{.preorder?(stronger} \\
\{-2\} & \text{strongPartialRefinement}(\text{pl1}, \text{pl2}, s) \\
\{-3\} & \text{strongerPLrefinement}(\text{pl2}, \text{pl4}) \\
\hline
\{1\} & \text{strongerPLrefinement}(\text{pl1}, \text{pl1})
\end{array}$$

Expanding the definition of preorder?,

`commutableDiagram2.2:`

{-1}	reflexive?(strongerPLrefinement) & transitive?(strongerPLrefinement)
{-2}	strongPartialRefinement(pl1, pl2, s)
{-3}	strongerPLrefinement(pl2, pl4)
{1}	strongerPLrefinement(pl1, pl1)

Applying disjunctive simplification to flatten sequent,
`commutableDiagram2.2:`

{-1}	reflexive?(strongerPLrefinement)
{-2}	transitive?(strongerPLrefinement)
{-3}	strongPartialRefinement(pl1, pl2, s)
{-4}	strongerPLrefinement(pl2, pl4)
{1}	strongerPLrefinement(pl1, pl1)

Expanding the definition of reflexive?,
`commutableDiagram2.2:`

{-1}	FORALL (x: PL[Conf, FM, Asset, AssetName, CK, {——}, [——]]): strongerPLrefinement(x, x)
{-2}	transitive?(strongerPLrefinement)
{-3}	strongPartialRefinement(pl1, pl2, s)
{-4}	strongerPLrefinement(pl2, pl4)
{1}	strongerPLrefinement(pl1, pl1)

Instantiating the top quantifier in -1 with the terms: pl1,
This completes the proof of `commutableDiagram2.2`.
Q.E.D.