Verbose proof for `strongPartRefReflexive`.

`strongPartRefReflexive:`

|-------

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

`strongPartRefReflexive:`

|-------

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

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

`strongPartRefReflexive:`

|-------

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

Expanding the definition of strongPartialRefinement,

`strongPartRefReflexive:`

|-------

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

Applying bddsimp,

`strongPartRefReflexive:`

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

|-------

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

Using lemma assetRefinement,

`strongPartRefReflexive:`

{-1} orders[set[Asset]].preorder?(—-)

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

|-------

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

Expanding the definition of preorder?,

`strongPartRefReflexive:`

{-1} reflexive?(—-) & transitive?(—-)

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

|-------

{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`:

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

`strongPartRefTransitive`:

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

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

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

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

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

Expanding the definition of member,

`strongPartRefTransitive:`

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

Expanding the definition of subset?,
`strongPartRefTransitive:`

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

Expanding the definition of member,

`strongPartRefTransitive`:

{1}	$(((\text{FORALL } (x\colon \text{Conf})\colon s(x) \Rightarrow \{\text{——}\}(F(\text{pl1}))(x)) \wedge$
	$(\text{FORALL } (x\colon \text{Conf})\colon s(x) \Rightarrow \{\text{——}\}(F(\text{pl2}))(x)) \wedge$
	$(\text{FORALL } c\colon s(c) \Rightarrow (\text{prod}(\text{pl1}, c) \text{ —- } \text{prod}(\text{pl2}, c))))$
	$\wedge$
	$(\text{FORALL } (x\colon \text{Conf})\colon t(x) \Rightarrow \{\text{——}\}(F(\text{pl2}))(x)) \wedge$
	$(\text{FORALL } (x\colon \text{Conf})\colon t(x) \Rightarrow \{\text{——}\}(F(\text{pl3}))(x)) \wedge$
	$(\text{FORALL } c\colon t(c) \Rightarrow (\text{prod}(\text{pl2}, c) \text{ —- } \text{prod}(\text{pl3}, c))))$
	$\Rightarrow$
	$(\text{FORALL } (\text{x\_1}\colon \text{Conf})\colon s(\text{x\_1}) \wedge t(\text{x\_1}) \Rightarrow \{\text{——}\}(F(\text{pl1}))(\text{x\_1})) \wedge$
	$(\text{FORALL } (\text{x\_1}\colon \text{Conf})\colon s(\text{x\_1}) \wedge t(\text{x\_1}) \Rightarrow \{\text{——}\}(F(\text{pl3}))(\text{x\_1})) \wedge$
	$(\text{FORALL } c\colon s(c) \wedge t(c) \Rightarrow (\text{prod}(\text{pl1}, c) \text{ —- } \text{prod}(\text{pl3}, c)))$

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

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

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

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

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

`strongPartRefTransitive.1:`

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

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

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

Using lemma assetRefinement,
`strongPartRefTransitive.1:`

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

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

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

Applying disjunctive simplification to flatten sequent,

`strongPartRefTransitive.1:`

{-1}	reflexive?(—-)
{-2}	transitive?(—-)
{-3}	$s(c)$ ⇒ {——}(F(pl1))(c)
{-4}	$s(c)$ ⇒ {——}(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}	$s(c)$
{-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) ⇒ (x —- z)
{-3}	$s(c)$ ⇒ {——}(F(pl1))(c)
{-4}	$s(c)$ ⇒ {——}(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}	$s(c)$
{-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}	$s(c) \wedge t(c)$ ⇒ {——}(F(pl3))(c)

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

`strongPartRefTransitive.2:`

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

Applying bddsimp,
`strongPartRefTransitive.2:`

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

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

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

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

`strongPartRefTransitive.3`:

{-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(pl1))(x_1)

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

{-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(pl1))(c)

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

{-1}	$s(c)$ ⇒ {——}(F(pl1))(c)
{-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(pl1))(c)

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

Verbose proof for `fmCompStrongDef`.

`fmCompStrongDef`:

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

`fmCompStrongDef`:

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

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

`fmCompStrongDef`:

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

Expanding the definition of subset?,

`fmCompStrongDef`:

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

Expanding the definition of member,

`fmCompStrongDef`:

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

Expanding the definition of prod,
`fmCompStrongDef`:

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

Using lemma assetRefinement,
`fmCompStrongDef`:

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

Expanding the definition of preorder?,

`fmCompStrongDef`:

{-1} reflexive?(—-) & transitive?(—-)

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

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

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

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

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

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

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

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

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

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

`fmCompStrongDef.2`:

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

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

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

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

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

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

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

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

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

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

`fmCompStrongDef.3`:

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

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

Verbose proof for `partPlusTotalImpliesPartFun`.
`partPlusTotalImpliesPartFun`:

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

`partPlusTotalImpliesPartFun`:

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

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

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

Applying bddsimp,
`partPlusTotalImpliesPartFun`:

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

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`:

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

Applying bddsimp,

`partPlusTotalImpliesPartFun`:

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

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

`partPlusTotalImpliesPartFun`:

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

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

`partPlusTotalImpliesPartFun`:

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

Applying bddsimp,

`partPlusTotalImpliesPartFun`:

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

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

`partPlusTotalImpliesPartFun`:

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

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

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

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

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

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

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

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

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

$$
\begin{array}{r|l}
\{-1\} & \text{reflexive?}(\text{—-}) \\
\{-2\} & \textsc{forall}\ (x\text{: set}[\text{Asset}]),\ (y\text{: set}[\text{Asset}]),\ (z\text{: set}[\text{Asset}]): \\
 & \quad (x \text{ —- } y)\ \&\ (y \text{ —- } z) \Rightarrow (x \text{ —- } z) \\
\{-3\} & \text{strongPartialRefinement}(\text{pl1},\ \text{pl2},\ s) \\
\{-4\} & s(c) \\
\{-5\} & (\{\text{——}\}(F(\text{pl2}))(f(c))) \\
\{-6\} & (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl2},\ f(c))) \\
\{-7\} & \text{plRefinement}(\text{pl2},\ \text{pl3}) \\
\{-8\} & (\text{prod}(\text{pl2},\ f(c)) \text{ —- } \text{prod}(\text{pl3},\ g(f(c)))) \\
\hline
\{1\} & (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl3},\ g(f(c))))
\end{array}
$$

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

$$
\begin{array}{r|l}
\{-1\} & \text{reflexive?}(\text{—-}) \\
\{-2\} & (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl2},\ f(c)))\ \& \\
 & \ (\text{prod}(\text{pl2},\ f(c)) \text{ —- } \text{prod}(\text{pl3},\ g(f(c)))) \\
 & \ \Rightarrow (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl3},\ g(f(c)))) \\
\{-3\} & \text{strongPartialRefinement}(\text{pl1},\ \text{pl2},\ s) \\
\{-4\} & s(c) \\
\{-5\} & (\{\text{——}\}(F(\text{pl2}))(f(c))) \\
\{-6\} & (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl2},\ f(c))) \\
\{-7\} & \text{plRefinement}(\text{pl2},\ \text{pl3}) \\
\{-8\} & (\text{prod}(\text{pl2},\ f(c)) \text{ —- } \text{prod}(\text{pl3},\ g(f(c)))) \\
\hline
\{1\} & (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl3},\ g(f(c))))
\end{array}
$$

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

$$
\begin{array}{r|l}
\{-1\} & \text{strongPartialRefinement}(\text{pl1},\ \text{pl2},\ s) \\
\{-2\} & \textsc{forall}\ c: \\
 & \quad s(c) \Rightarrow (\{\text{——}\}(F(\text{pl2}))(f(c))) \wedge (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl2},\ f(c))) \\
\{-3\} & \text{plRefinement}(\text{pl2},\ \text{pl3}) \\
\{-4\} & \text{plRefinementFun}(\text{pl2},\ \text{pl3},\ g) \\
\hline
\{1\} & \forall\ (x_1\text{: }(s))\text{: } \{\text{——}\}(F(\text{pl2}))(f(x_1))
\end{array}
$$

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

`partPlusTotalImpliesPartFun.2:`

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

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

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

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

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

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

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

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

```
partPlusTotalImpliesPartFun.2:
```

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

Expanding the definition of member,

```
partPlusTotalImpliesPartFun.2:
```

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

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

```
partPlusTotalImpliesPartFun.2:
```

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

Applying bddsimp,

```
partPlusTotalImpliesPartFun.2:
```

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

Adding type constraints for c,

`partPlusTotalImpliesPartFun.2`:

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

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

$\{1\}$ $(\text{FORALL } (x\text{: Conf}): (x \in s) \Rightarrow (x \in \{\text{——}\}(F(\text{pl1})))) \wedge$
$(\text{FORALL } (x\text{: Conf}): (x \in s) \Rightarrow (x \in \{\text{——}\}(F(\text{pl3})))) \wedge$
$(\text{FORALL } c\text{: } s(c) \Rightarrow (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl3},\ c)))$

Expanding the definition of member,
`partPlusTotalStrongerImpliesPart:`

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

$\{1\}$ $(\text{FORALL } (x\text{: Conf}): s(x) \Rightarrow \{\text{——}\}(F(\text{pl1}))(x)) \wedge$
$(\text{FORALL } (x\text{: Conf}): s(x) \Rightarrow \{\text{——}\}(F(\text{pl3}))(x)) \wedge$
$(\text{FORALL } c\text{: } s(c) \Rightarrow (\text{prod}(\text{pl1},\ c) \text{ —- } \text{prod}(\text{pl3},\ c)))$

Expanding the definition of prod,
`partPlusTotalStrongerImpliesPart:`

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

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

Applying bddsimp,
we get 2 subgoals:

`partPlusTotalStrongerImpliesPart.1:`

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

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

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

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

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

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

`partPlusTotalStrongerImpliesPart.1`:

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

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

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

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

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

Using lemma assetRefinement,
`partPlusTotalStrongerImpliesPart.1`:

{-1}   orders[set[Asset]].preorder?(—-)
{-2} $s(c) \Rightarrow \{——\}(F(\mathrm{pl1}))(c)$
{-3} $s(c) \Rightarrow \{——\}(F(\mathrm{pl2}))(c)$
{-4}   FORALL c:
  $s(c) \Rightarrow$
   $(([——](K(\mathrm{pl1}))(A(\mathrm{pl1}))(c))$ —- $([——](K(\mathrm{pl2}))(A(\mathrm{pl2}))(c)))$
{-5} $\{——\}(F(\mathrm{pl2}))(c) \Rightarrow$
  $(\{——\}(F(\mathrm{pl3}))(c) \wedge$
   $(([——](K(\mathrm{pl2}))(A(\mathrm{pl2}))(c))$ —- $([——](K(\mathrm{pl3}))(A(\mathrm{pl3}))(c))))$

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

Expanding the definition of preorder?,

`partPlusTotalStrongerImpliesPart.1:`

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

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

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

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

$$
\begin{array}{l|ll}
& \{\text{-}1\} & \text{reflexive?}(\text{—-}) \\
& \{\text{-}2\} & \textsc{forall}\ (x\text{: set}[\text{Asset}]),\ (y\text{: set}[\text{Asset}]),\ (z\text{: set}[\text{Asset}]): \\
& & \quad (x \;\text{—-}\; y) \;\&\; (y \;\text{—-}\; z) \Rightarrow (x \;\text{—-}\; z) \\
& \{\text{-}3\} & s(c) \Rightarrow \{\text{——}\}(F(\text{pl1}))(c) \\
& \{\text{-}4\} & s(c) \Rightarrow \{\text{——}\}(F(\text{pl2}))(c) \\
& \{\text{-}5\} & \textsc{forall}\ c: \\
& & \quad s(c) \Rightarrow \\
& & \quad\quad (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c)) \;\text{—-}\; ([\text{——}](K(\text{pl2}))(A(\text{pl2}))(c))) \\
& \{\text{-}6\} & \{\text{——}\}(F(\text{pl2}))(c) \Rightarrow \\
& & \quad (\{\text{——}\}(F(\text{pl3}))(c) \wedge \\
& & \quad\quad (([\text{——}](K(\text{pl2}))(A(\text{pl2}))(c)) \;\text{—-}\; ([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c)))) \\
& \{\text{-}7\} & s(c) \\
\hline
& \{1\} & (([\text{——}](K(\text{pl1}))(A(\text{pl1}))(c)) \;\text{—-}\; ([\text{——}](K(\text{pl3}))(A(\text{pl3}))(c)))
\end{array}
$$

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`:

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

Applying bddsimp,

`partPlusTotalStrongerImpliesPart.1`:

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

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

`partPlusTotalStrongerImpliesPart.1`:

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

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

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

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

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

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

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

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

`partPlusTotalStrongerImpliesPart.2`:

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

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

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

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

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

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 (c1: Conf):
        {——}(F(pl1))(c1) ⇒
         ({——}(F(pl3))(c1) ∧
            ((([——](K(pl1))(A(pl1))(c1)) —- ([——](K(pl3))(A(pl3))(c1))))
{-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 (c1: Conf):
           {——}(F(pl4))(c1) ⇒
            ({——}(F(pl4))(c1) ∧
               ((([——](K(pl4))(A(pl4))(c1)) —-
                  ([——](K(pl4))(A(pl4))(c1)))))
```

Expanding the definition of subset?,
`commutableDiagram`:

```
{-1}  FORALL (c1: Conf):
        {——}(F(pl1))(c1) ⇒
         ({——}(F(pl3))(c1) ∧
            ((([——](K(pl1))(A(pl1))(c1)) —- ([——](K(pl3))(A(pl3))(c1))))
{-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 (c1: Conf):
           {——}(F(pl4))(c1) ⇒
            ({——}(F(pl4))(c1) ∧
               ((([——](K(pl4))(A(pl4))(c1)) —-
                  ([——](K(pl4))(A(pl4))(c1)))))
```

Expanding the definition of member,

`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): s(x) ⇒ {——}(F(pl3))(x)) ∧
       (FORALL (x: Conf): s(x) ⇒ {——}(F(pl4))(x)) ∧
        (FORALL c: s(c) ⇒ (prod(pl3, c) —- prod(pl4, c)))
--------------------------------------------------------------------------
{1}   ((FORALL (x: Conf): s(x) ⇒ {——}(F(pl4))(x)) ∧
         (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)))))
```

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

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

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

`commutableDiagram.1:`

{-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): $s(x)$ ⇒ {——}(F(pl3))(x)
{-3} FORALL (x: Conf): $s(x)$ ⇒ {——}(F(pl4))(x)
{-4} FORALL c: $s(c)$ ⇒ (prod(pl3, c) —- prod(pl4, c))

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

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

{1} {——}(F(pl4))(c) ⇒
  ({——}(F(pl4))(c) ∧
   (([——](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) $\Rightarrow$
({——}(F(pl3))(c_1) $\wedge$
(([——](K(pl1))(A(pl1))(c_1)) —- ([——](K(pl3))(A(pl3))(c_1))))
{-4} FORALL (x: Conf): $s(x)$ $\Rightarrow$ {——}(F(pl3))(x)
{-5} FORALL (x: Conf): $s(x)$ $\Rightarrow$ {——}(F(pl4))(x)
{-6} FORALL c: $s(c)$ $\Rightarrow$ (prod(pl3, c) —- prod(pl4, c))
{-7} {——}(F(pl4))(c)

{1} ({——}(F(pl4))(c) $\wedge$
(([——](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) $\Rightarrow$
({——}(F(pl3))(c_1) $\wedge$
(([——](K(pl1))(A(pl1))(c_1)) —- ([——](K(pl3))(A(pl3))(c_1))))
{-4} FORALL (x: Conf): $s(x)$ $\Rightarrow$ {——}(F(pl3))(x)
{-5} FORALL (x: Conf): $s(x)$ $\Rightarrow$ {——}(F(pl4))(x)
{-6} FORALL c: $s(c)$ $\Rightarrow$ (prod(pl3, c) —- prod(pl4, c))
{-7} {——}(F(pl4))(c)

{1} ({——}(F(pl4))(c) $\wedge$
(([——](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) $\Rightarrow$
({——}(F(pl3))(c_1) $\wedge$
(([——](K(pl1))(A(pl1))(c_1)) —- ([——](K(pl3))(A(pl3))(c_1))))
{-4} FORALL (x: Conf): $s(x)$ $\Rightarrow$ {——}(F(pl3))(x)
{-5} FORALL (x: Conf): $s(x)$ $\Rightarrow$ {——}(F(pl4))(x)
{-6} FORALL c: $s(c)$ $\Rightarrow$ (prod(pl3, c) —- prod(pl4, c))
{-7} {——}(F(pl4))(c)

{1} ({——}(F(pl4))(c) $\wedge$
(([——](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`:

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

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

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

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

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

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

`commutableDiagram.2:`

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

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

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

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

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

Using lemma assetRefinement,
`commutableDiagram.2:`

{-1} orders[set[Asset]].preorder?(—-)
{-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)))

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} (([——](K(pl1))(A(pl1))(c)) —- ([——](K(pl3))(A(pl3))(c))) &
  (([——](K(pl3))(A(pl3))(c)) —- ([——](K(pl4))(A(pl4))(c)))
  ⇒ (([——](K(pl1))(A(pl1))(c)) —- ([——](K(pl4))(A(pl4))(c)))
{-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)))

Applying bddsimp,
`commutableDiagram.2:`

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

{1} (([——](K(pl1))(A(pl1))(c)) —- ([——](K(pl3))(A(pl3))(c)))
{2} (([——](K(pl1))(A(pl1))(c)) —- ([——](K(pl4))(A(pl4))(c)))
{3} {——}(F(pl1))(c)

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

{-1} reflexive?(—-)
{-2} (([——](K(pl3))(A(pl3))(c)) —- ([——](K(pl4))(A(pl4))(c)))
{-3} {——}(F(pl3))(c)
{-4} s(c)
{-5} s(c) ⇒ {——}(F(pl4))(c)

{1} (([——](K(pl1))(A(pl1))(c)) —- ([——](K(pl3))(A(pl3))(c)))
{2} (([——](K(pl1))(A(pl1))(c)) —- ([——](K(pl4))(A(pl4))(c)))
{3} {——}(F(pl1))(c)

Adding type constraints for s,

`commutableDiagram.2:`

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

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

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

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

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

Expanding the definition of member,

`commutableDiagram.2`:

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

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:`

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

Applying totalImpliesPartial
`commutableDiagram2.1:`

$$
\begin{array}{|ll}
\{-1\} & \forall\ (\text{pl1},\ \text{pl2},\ s\text{: set}[\text{Conf}]\ \mid\ (s \subseteq \{\text{——}\}(F(\text{pl1})))): \\
 & \quad \text{strongerPLrefinement}(\text{pl1},\ \text{pl2}) \Rightarrow \text{strongPartialRefinement}(\text{pl1},\ \text{pl2},\ s) \\
\{-2\} & \text{strongPartialRefinement}(\text{pl1},\ \text{pl2},\ s) \\
\{-3\} & \text{strongerPLrefinement}(\text{pl2},\ \text{pl4}) \\
\hline
\{1\} & \text{strongPartialRefinement}(\text{pl1},\ \text{pl4},\ s)
\end{array}
$$

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

$$
\begin{array}{|ll}
\{-1\} & \text{strongerPLrefinement}(\text{pl2},\ \text{pl4}) \Rightarrow \text{strongPartialRefinement}(\text{pl2},\ \text{pl4},\ s) \\
\{-2\} & \text{strongPartialRefinement}(\text{pl1},\ \text{pl2},\ s) \\
\{-3\} & \text{strongerPLrefinement}(\text{pl2},\ \text{pl4}) \\
\hline
\{1\} & \text{strongPartialRefinement}(\text{pl1},\ \text{pl4},\ s)
\end{array}
$$

Applying bddsimp,
`commutableDiagram2.1.1:`

$$
\begin{array}{|ll}
\{-1\} & \text{strongerPLrefinement}(\text{pl2},\ \text{pl4}) \\
\{-2\} & \text{strongPartialRefinement}(\text{pl2},\ \text{pl4},\ s) \\
\{-3\} & \text{strongPartialRefinement}(\text{pl1},\ \text{pl2},\ s) \\
\hline
\{1\} & \text{strongPartialRefinement}(\text{pl1},\ \text{pl4},\ s)
\end{array}
$$

Applying strongPartRefTransitive
`commutableDiagram2.1.1:`

$$
\begin{array}{|ll}
\{-1\} & \forall\ (\text{pl1},\ \text{pl2},\ \text{pl3},\ s,\ t): \\
 & \quad (\text{strongPartialRefinement}(\text{pl1},\ \text{pl2},\ s) \wedge \text{strongPartialRefine-} \\
 & \text{ment}(\text{pl2},\ \text{pl3},\ t)) \Rightarrow \\
 & \qquad \text{strongPartialRefinement}(\text{pl1},\ \text{pl3},\ (s \cap t)) \\
\{-2\} & \text{strongerPLrefinement}(\text{pl2},\ \text{pl4}) \\
\{-3\} & \text{strongPartialRefinement}(\text{pl2},\ \text{pl4},\ s) \\
\{-4\} & \text{strongPartialRefinement}(\text{pl1},\ \text{pl2},\ s) \\
\hline
\{1\} & \text{strongPartialRefinement}(\text{pl1},\ \text{pl4},\ s)
\end{array}
$$

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

{-1}	(strongPartialRefinement(pl1, pl2, s) $\wedge$ strongPartialRefinement(pl2, pl4, s)) $\Rightarrow$ 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}	$\forall$ (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:`

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

{1} strongPartialRefinement(pl1, pl4, s)

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

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

{1} $(s \subseteq \{\text{——}\}(F(\text{pl2})))$
{2} strongPartialRefinement(pl1, pl4, s)

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

{-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} strongerPLrefinement(pl2, pl4)

{1} $(s \subseteq \{\text{——}\}(F(\text{pl2})))$
{2} $(s \subseteq \{\text{——}\}(F(\text{pl1}))) \wedge$ $(s \subseteq \{\text{——}\}(F(\text{pl4}))) \wedge$ (FORALL c: $s(c) \Rightarrow$ (prod(pl1, c) —- prod(pl4, c)))

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

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

{1} strongerPLrefinement(pl1, pl1)

Applying strongerPLref
`commutableDiagram2.2:`

{-1} orders[PL[Conf, FM, Asset, AssetName, CK, {——}, [——]]].preorder?(stronge
{-2} strongPartialRefinement(pl1, pl2, s)
{-3} strongerPLrefinement(pl2, pl4)

{1} strongerPLrefinement(pl1, pl1)

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.

Verbose proof for `changeAssetStrongPartialRef`.
`changeAssetStrongPartialRef`:

$$
\begin{array}{|l}
\\ \hline
\{1\} \quad \text{FORALL } (\text{pl}, \text{ am2}, \text{ pairs}, a_1, a_2, \text{ an}, s): \\
\qquad ((\text{syntaxChangeAsset}(A(\text{pl}), \text{ am2}, \text{ pairs}, a_1, a_2, \text{ an}) \wedge \\
\qquad\qquad s = \diamond\ (F(\text{pl}), K(\text{pl}), \text{ singleton}(\text{an}))) \\
\qquad\quad \Rightarrow \text{strongPartialRefinement}(\text{pl}, \text{ pl2}, s)) \\
\qquad\quad \text{WHERE pl2} = (\#F := F(\text{pl}), A := \text{am2}, K := K(\text{pl})\#)
\end{array}
$$

`changeAssetStrongPartialRef`:

$$
\begin{array}{|l}
\\ \hline
\{1\} \quad \text{FORALL } (\text{pl}, \text{ am2}, \text{ pairs}, a_1, a_2, \text{ an}, s): \\
\qquad ((\text{syntaxChangeAsset}(A(\text{pl}), \text{ am2}, \text{ pairs}, a_1, a_2, \text{ an}) \wedge \\
\qquad\qquad s = \diamond\ (F(\text{pl}), K(\text{pl}), \text{ singleton}(\text{an}))) \\
\qquad\quad \Rightarrow \text{strongPartialRefinement}(\text{pl}, \text{ pl2}, s)) \\
\qquad\quad \text{WHERE pl2} = (\#F := F(\text{pl}), A := \text{am2}, K := K(\text{pl})\#)
\end{array}
$$

For the top quantifier in 1, we introduce Skolem constants: (pl am2 pairs a1 a2 an s),
`changeAssetStrongPartialRef`:

$$
\begin{array}{|l}
\\ \hline
\{1\} \quad ((\text{syntaxChangeAsset}(A(\text{pl}), \text{ am2}, \text{ pairs}, a_1, a_2, \text{ an}) \wedge \\
\qquad\qquad s = (\diamond)(F(\text{pl}), K(\text{pl}), \text{ singleton}(\text{an}))) \\
\qquad\quad \Rightarrow \text{strongPartialRefinement}(\text{pl}, \text{ pl2}, s)) \\
\qquad \text{WHERE pl2} = (\#F := F(\text{pl}), A := \text{am2}, K := K(\text{pl})\#)
\end{array}
$$

Expanding the definition of strongPartialRefinement,
`changeAssetStrongPartialRef`:

$$
\begin{array}{|l}
\\ \hline
\{1\} \quad ((\text{syntaxChangeAsset}(A(\text{pl}), \text{ am2}, \text{ pairs}, a_1, a_2, \text{ an}) \wedge \\
\qquad\qquad s = (\diamond)(F(\text{pl}), K(\text{pl}), \text{ singleton}(\text{an}))) \\
\qquad \Rightarrow \\
\qquad (s \subseteq [\text{——}](F(\text{pl}))) \wedge \\
\qquad (s \subseteq [\text{——}](F(\text{pl}))) \wedge \\
\qquad\quad (\text{FORALL } (c: \text{ Configuration}): \\
\qquad\qquad s(c) \Rightarrow \\
\qquad\qquad\quad (\text{prod}(\text{pl}, c) \text{ —- } \text{prod}((\#F := F(\text{pl}), A := \text{am2}, K := K(\text{pl})\#), c)))
\end{array}
$$

Applying bddsimp,
we get 2 subgoals:

`changeAssetStrongPartialRef.1:`

$$
\begin{array}{l|ll}
& \{-1\} & \text{syntaxChangeAsset}(A(\text{pl}),\ \text{am2},\ \text{pairs},\ a_1,\ a_2,\ \text{an}) \\
& \{-2\} & s\ =\ (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an})) \\
\hline
& \{1\} & \textsc{forall}\ (c\colon\ \text{Configuration})\colon \\
& & \quad s(c)\ \Rightarrow\ (\text{prod}(\text{pl},\ c)\ \text{—-}\ \text{prod}((\#F\ :=\ F(\text{pl}),\ A\ :=\ \text{am2},\ K\ :=\ K(\text{pl})\#),\ c))
\end{array}
$$

Applying sameEvalPairs

`changeAssetStrongPartialRef.1:`

$$
\begin{array}{l|ll}
& \{-1\} & \forall\ (\text{fm}\colon\ \text{FMi},\ \text{am}, \\
& & \qquad \text{ck}\colon \\
& & \qquad\quad \text{CK} \\
& & \qquad\qquad \big[\text{Configuration},\ \text{FeatureExpression},\ \text{sat},\ \text{FMi},\ \text{Fea-} \\
& & \text{ture},\ [\text{——}],\ \text{wf},\ \text{wt}, \\
& & \qquad\qquad\quad \text{genFeatureExpression},\ \text{getFeatures},\ \text{addMandatory},\ \text{ad-} \\
& & \text{dOptional}\big], \\
& & \qquad \text{am2},\ \text{pairs},\ a_1,\ a_2,\ \text{an},\ s\colon\ \text{set}\big[\text{Configuration}\big])\colon \\
& & \quad ((\text{syntaxChangeAsset}(\text{am},\ \text{am2},\ \text{pairs},\ a_1,\ a_2,\ \text{an})\ \wedge\ s\ =\ (\Diamond)(\text{fm},\ \text{ck},\ \text{sin-} \\
& & \text{gleton}(\text{an}))) \\
& & \qquad \Rightarrow \\
& & \qquad (\textsc{forall}\ (c\colon\ \text{Configuration})\colon \\
& & \qquad\quad s(c)\ \Rightarrow\ (\text{semantics}(\text{ck})(\text{am})(c))\ =\ \text{semantics}(\text{ck})(\text{pairs})(c))) \\
& \{-2\} & \text{syntaxChangeAsset}(A(\text{pl}),\ \text{am2},\ \text{pairs},\ a_1,\ a_2,\ \text{an}) \\
& \{-3\} & s\ =\ (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an})) \\
\hline
& \{1\} & \textsc{forall}\ (c\colon\ \text{Configuration})\colon \\
& & \quad s(c)\ \Rightarrow\ (\text{prod}(\text{pl},\ c)\ \text{—-}\ \text{prod}((\#F\ :=\ F(\text{pl}),\ A\ :=\ \text{am2},\ K\ :=\ K(\text{pl})\#),\ c))
\end{array}
$$

Instantiating the top quantifier in -1 with the terms: F(pl), A(pl), K(pl), am2, pairs, a1, a2, an, s,

`changeAssetStrongPartialRef.1:`

$$
\begin{array}{l|ll}
& \{-1\} & ((\text{syntaxChangeAsset}(A(\text{pl}),\ \text{am2},\ \text{pairs},\ a_1,\ a_2,\ \text{an})\ \wedge \\
& & \qquad s\ =\ (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an}))) \\
& & \quad \Rightarrow \\
& & \quad (\textsc{forall}\ (c\colon\ \text{Configuration})\colon \\
& & \qquad s(c)\ \Rightarrow \\
& & \qquad\ (\text{semantics}(K(\text{pl}))(A(\text{pl}))(c))\ =\ \text{semantics}(K(\text{pl}))(\text{pairs})(c))) \\
& \{-2\} & \text{syntaxChangeAsset}(A(\text{pl}),\ \text{am2},\ \text{pairs},\ a_1,\ a_2,\ \text{an}) \\
& \{-3\} & s\ =\ (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an})) \\
\hline
& \{1\} & \textsc{forall}\ (c\colon\ \text{Configuration})\colon \\
& & \quad s(c)\ \Rightarrow\ (\text{prod}(\text{pl},\ c)\ \text{—-}\ \text{prod}((\#F\ :=\ F(\text{pl}),\ A\ :=\ \text{am2},\ K\ :=\ K(\text{pl})\#),\ c))
\end{array}
$$

Applying bddsimp,

`changeAssetStrongPartialRef.1:`

{-1} syntaxChangeAsset(A(pl), am2, pairs, a_1, a_2, an)
{-2} s = (◇)(F(pl), K(pl), singleton(an))
{-3} FORALL (c: Configuration):
  $s(c)$ ⇒ (semantics(K(pl))(A(pl))(c)) = semantics(K(pl))(pairs)(c)

{1} FORALL (c: Configuration):
  $s(c)$ ⇒ (prod(pl, c) —- prod((#F := F(pl), A := am2, K := K(pl)#), c))

Applying sameEvalPairs2
`changeAssetStrongPartialRef.1:`

{-1} ∀ (fm: FMi, am,
  ck:
   CK
    [Configuration, FeatureExpression, sat, FMi, Feature, [——], wf, wt,
     genFeatureExpression, getFeatures, addMandatory, addOptional],
  am2, pairs, a_1, a_2, an, s: set[Configuration]):
 ((syntaxChangeAsset(am, am2, pairs, a_1, a_2, an) ∧ s = (◇)(fm, ck, singleton(an)))
  ⇒
  (FORALL (c: Configuration):
   $s(c)$ ⇒ (semantics(ck)(am2)(c)) = semantics(ck)(pairs)(c)))
{-2} syntaxChangeAsset(A(pl), am2, pairs, a_1, a_2, an)
{-3} s = (◇)(F(pl), K(pl), singleton(an))
{-4} FORALL (c: Configuration):
  $s(c)$ ⇒ (semantics(K(pl))(A(pl))(c)) = semantics(K(pl))(pairs)(c)

{1} FORALL (c: Configuration):
  $s(c)$ ⇒ (prod(pl, c) —- prod((#F := F(pl), A := am2, K := K(pl)#), c))

Instantiating the top quantifier in -1 with the terms: F(pl), A(pl), K(pl), am2, pairs, a1, a2, an, s,

`changeAssetStrongPartialRef.1:`

$$
\begin{array}{l|ll}
 & \{-1\} & ((\text{syntaxChangeAsset}(A(\text{pl}),\ \text{am2},\ \text{pairs},\ a_1,\ a_2,\ \text{an})\ \wedge \\
 & & \qquad s = (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an}))) \\
 & & \quad \Rightarrow \\
 & & \quad (\text{FORALL}\ (c\text{: Configuration}): \\
 & & \qquad s(c) \Rightarrow (\text{semantics}(K(\text{pl}))(\text{am2})(c)) = \text{semantics}(K(\text{pl}))(\text{pairs})(c))) \\
 & \{-2\} & \text{syntaxChangeAsset}(A(\text{pl}),\ \text{am2},\ \text{pairs},\ a_1,\ a_2,\ \text{an}) \\
 & \{-3\} & s = (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an})) \\
 & \{-4\} & \text{FORALL}\ (c\text{: Configuration}): \\
 & & \quad s(c) \Rightarrow (\text{semantics}(K(\text{pl}))(A(\text{pl}))(c)) = \text{semantics}(K(\text{pl}))(\text{pairs})(c) \\
\hline
 & \{1\} & \text{FORALL}\ (c\text{: Configuration}): \\
 & & \quad s(c) \Rightarrow (\text{prod}(\text{pl},\ c) \text{ —- } \text{prod}((\#F := F(\text{pl}),\ A := \text{am2},\ K := K(\text{pl})\#),\ c))
\end{array}
$$

Applying bddsimp,

`changeAssetStrongPartialRef.1:`

$$
\begin{array}{l|ll}
 & \{-1\} & \text{syntaxChangeAsset}(A(\text{pl}),\ \text{am2},\ \text{pairs},\ a_1,\ a_2,\ \text{an}) \\
 & \{-2\} & s = (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an})) \\
 & \{-3\} & \text{FORALL}\ (c\text{: Configuration}): \\
 & & \quad s(c) \Rightarrow (\text{semantics}(K(\text{pl}))(\text{am2})(c)) = \text{semantics}(K(\text{pl}))(\text{pairs})(c) \\
 & \{-4\} & \text{FORALL}\ (c\text{: Configuration}): \\
 & & \quad s(c) \Rightarrow (\text{semantics}(K(\text{pl}))(A(\text{pl}))(c)) = \text{semantics}(K(\text{pl}))(\text{pairs})(c) \\
\hline
 & \{1\} & \text{FORALL}\ (c\text{: Configuration}): \\
 & & \quad s(c) \Rightarrow (\text{prod}(\text{pl},\ c) \text{ —- } \text{prod}((\#F := F(\text{pl}),\ A := \text{am2},\ K := K(\text{pl})\#),\ c))
\end{array}
$$

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

`changeAssetStrongPartialRef.1:`

$$
\begin{array}{l|ll}
 & \{-1\} & \text{syntaxChangeAsset}(A(\text{pl}),\ \text{am2},\ \text{pairs},\ a_1,\ a_2,\ \text{an}) \\
 & \{-2\} & s = (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an})) \\
 & \{-3\} & \text{FORALL}\ (c\text{: Configuration}): \\
 & & \quad s(c) \Rightarrow (\text{semantics}(K(\text{pl}))(\text{am2})(c)) = \text{semantics}(K(\text{pl}))(\text{pairs})(c) \\
 & \{-4\} & \text{FORALL}\ (c\text{: Configuration}): \\
 & & \quad s(c) \Rightarrow (\text{semantics}(K(\text{pl}))(A(\text{pl}))(c)) = \text{semantics}(K(\text{pl}))(\text{pairs})(c) \\
\hline
 & \{1\} & s(c) \Rightarrow (\text{prod}(\text{pl},\ c) \text{ —- } \text{prod}((\#F := F(\text{pl}),\ A := \text{am2},\ K := K(\text{pl})\#),\ c))
\end{array}
$$

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

`changeAssetStrongPartialRef.1:`

$$
\begin{array}{l|ll}
 & \{-1\} & \text{syntaxChangeAsset}(A(\text{pl}),\ \text{am2},\ \text{pairs},\ a_1,\ a_2,\ \text{an}) \\
 & \{-2\} & s = (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an})) \\
 & \{-3\} & s(c) \Rightarrow (\text{semantics}(K(\text{pl}))(\text{am2})(c)) = \text{semantics}(K(\text{pl}))(\text{pairs})(c) \\
 & \{-4\} & \text{FORALL}\ (c\text{: Configuration}): \\
 & & \quad s(c) \Rightarrow (\text{semantics}(K(\text{pl}))(A(\text{pl}))(c)) = \text{semantics}(K(\text{pl}))(\text{pairs})(c) \\
\hline
 & \{1\} & s(c) \Rightarrow (\text{prod}(\text{pl},\ c) \text{ —- } \text{prod}((\#F := F(\text{pl}),\ A := \text{am2},\ K := K(\text{pl})\#),\ c))
\end{array}
$$

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

`changeAssetStrongPartialRef.1:`

$$
\begin{array}{l|ll}
 & \{-1\} & \text{syntaxChangeAsset}(A(\text{pl}),\ \text{am2},\ \text{pairs},\ a_1,\ a_2,\ \text{an}) \\
 & \{-2\} & s = (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an})) \\
 & \{-3\} & s(c) \Rightarrow (\text{semantics}(K(\text{pl}))(\text{am2})(c)) = \text{semantics}(K(\text{pl}))(\text{pairs})(c) \\
 & \{-4\} & s(c) \Rightarrow (\text{semantics}(K(\text{pl}))(A(\text{pl}))(c)) = \text{semantics}(K(\text{pl}))(\text{pairs})(c) \\
\hline
 & \{1\} & s(c) \Rightarrow (\text{prod}(\text{pl},\ c) \text{ —- } \text{prod}((\#F := F(\text{pl}),\ A := \text{am2},\ K := K(\text{pl})\#),\ c))
\end{array}
$$

Expanding the definition of prod,
`changeAssetStrongPartialRef.1:`

$$
\begin{array}{l|ll}
 & \{-1\} & \text{syntaxChangeAsset}(A(\text{pl}),\ \text{am2},\ \text{pairs},\ a_1,\ a_2,\ \text{an}) \\
 & \{-2\} & s = (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an})) \\
 & \{-3\} & s(c) \Rightarrow (\text{semantics}(K(\text{pl}))(\text{am2})(c)) = \text{semantics}(K(\text{pl}))(\text{pairs})(c) \\
 & \{-4\} & s(c) \Rightarrow (\text{semantics}(K(\text{pl}))(A(\text{pl}))(c)) = \text{semantics}(K(\text{pl}))(\text{pairs})(c) \\
\hline
 & \{1\} & s(c) \Rightarrow \\
 & & \quad ((\text{semantics}(K(\text{pl}))(A(\text{pl}))(c)) \text{ —- } (\text{semantics}(K(\text{pl}))(\text{am2})(c)))
\end{array}
$$

Applying bddsimp,
`changeAssetStrongPartialRef.1:`

$$
\begin{array}{l|ll}
 & \{-1\} & \text{syntaxChangeAsset}(A(\text{pl}),\ \text{am2},\ \text{pairs},\ a_1,\ a_2,\ \text{an}) \\
 & \{-2\} & s = (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an})) \\
 & \{-3\} & s(c) \\
 & \{-4\} & (\text{semantics}(K(\text{pl}))(\text{am2})(c)) = \text{semantics}(K(\text{pl}))(\text{pairs})(c) \\
 & \{-5\} & (\text{semantics}(K(\text{pl}))(A(\text{pl}))(c)) = \text{semantics}(K(\text{pl}))(\text{pairs})(c) \\
\hline
 & \{1\} & ((\text{semantics}(K(\text{pl}))(A(\text{pl}))(c)) \text{ —- } (\text{semantics}(K(\text{pl}))(\text{am2})(c)))
\end{array}
$$

Replacing using formula -4,
`changeAssetStrongPartialRef.1:`

$$
\begin{array}{l|ll}
 & \{-1\} & \text{syntaxChangeAsset}(A(\text{pl}),\ \text{am2},\ \text{pairs},\ a_1,\ a_2,\ \text{an}) \\
 & \{-2\} & s = (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an})) \\
 & \{-3\} & s(c) \\
 & \{-4\} & (\text{semantics}(K(\text{pl}))(\text{am2})(c)) = \text{semantics}(K(\text{pl}))(\text{pairs})(c) \\
 & \{-5\} & (\text{semantics}(K(\text{pl}))(A(\text{pl}))(c)) = \text{semantics}(K(\text{pl}))(\text{pairs})(c) \\
\hline
 & \{1\} & ((\text{semantics}(K(\text{pl}))(A(\text{pl}))(c)) \text{ —- } \text{semantics}(K(\text{pl}))(\text{pairs})(c))
\end{array}
$$

Replacing using formula -5,
`changeAssetStrongPartialRef.1:`

$$
\begin{array}{l|ll}
 & \{-1\} & \text{syntaxChangeAsset}(A(\text{pl}),\ \text{am2},\ \text{pairs},\ a_1,\ a_2,\ \text{an}) \\
 & \{-2\} & s = (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an})) \\
 & \{-3\} & s(c) \\
 & \{-4\} & (\text{semantics}(K(\text{pl}))(\text{am2})(c)) = \text{semantics}(K(\text{pl}))(\text{pairs})(c) \\
 & \{-5\} & (\text{semantics}(K(\text{pl}))(A(\text{pl}))(c)) = \text{semantics}(K(\text{pl}))(\text{pairs})(c) \\
\hline
 & \{1\} & (\text{semantics}(K(\text{pl}))(\text{pairs})(c) \text{ —- } \text{semantics}(K(\text{pl}))(\text{pairs})(c))
\end{array}
$$

Using lemma SPLrefinement.assetRefinement,
`changeAssetStrongPartialRef.1:`

{-1} orders[set[Asset]].preorder?(—-)
{-2} syntaxChangeAsset(A(pl), am2, pairs, a_1, a_2, an)
{-3} s = (◇)(F(pl), K(pl), singleton(an))
{-4} $s(c)$
{-5} (semantics(K(pl))(am2)(c)) = semantics(K(pl))(pairs)(c)
{-6} (semantics(K(pl))(A(pl))(c)) = semantics(K(pl))(pairs)(c)

{1} (semantics(K(pl))(pairs)(c) —- semantics(K(pl))(pairs)(c))

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

{-1} reflexive?(—-) & transitive?(—-)
{-2} syntaxChangeAsset(A(pl), am2, pairs, a_1, a_2, an)
{-3} s = (◇)(F(pl), K(pl), singleton(an))
{-4} $s(c)$
{-5} (semantics(K(pl))(am2)(c)) = semantics(K(pl))(pairs)(c)
{-6} (semantics(K(pl))(A(pl))(c)) = semantics(K(pl))(pairs)(c)

{1} (semantics(K(pl))(pairs)(c) —- semantics(K(pl))(pairs)(c))

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

{-1} reflexive?(—-)
{-2} transitive?(—-)
{-3} syntaxChangeAsset(A(pl), am2, pairs, a_1, a_2, an)
{-4} s = (◇)(F(pl), K(pl), singleton(an))
{-5} $s(c)$
{-6} (semantics(K(pl))(am2)(c)) = semantics(K(pl))(pairs)(c)
{-7} (semantics(K(pl))(A(pl))(c)) = semantics(K(pl))(pairs)(c)

{1} (semantics(K(pl))(pairs)(c) —- semantics(K(pl))(pairs)(c))

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

{-1} FORALL (x: set[Asset]): (x —- x)
{-2} transitive?(—-)
{-3} syntaxChangeAsset(A(pl), am2, pairs, a_1, a_2, an)
{-4} s = (◇)(F(pl), K(pl), singleton(an))
{-5} $s(c)$
{-6} (semantics(K(pl))(am2)(c)) = semantics(K(pl))(pairs)(c)
{-7} (semantics(K(pl))(A(pl))(c)) = semantics(K(pl))(pairs)(c)

{1} (semantics(K(pl))(pairs)(c) —- semantics(K(pl))(pairs)(c))

Instantiating the top quantifier in -1 with the terms: semantics(K(pl))(pairs)(c),

This completes the proof of `changeAssetStrongPartialRef.1`.
`changeAssetStrongPartialRef.2`:

$$\begin{array}{|ll}
\{-1\} & \text{syntaxChangeAsset}(A(\text{pl}),\ \text{am2},\ \text{pairs},\ a_1,\ a_2,\ \text{an}) \\
\{-2\} & s\ =\ (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an})) \\ \hline
\{1\} & (s \subseteq [\text{——}](F(\text{pl})))
\end{array}$$

Applying filteredConfigurations
`changeAssetStrongPartialRef.2`:

$$\begin{array}{|ll}
\{-1\} & \forall\ (s\colon \text{set}\big[\text{Configuration}\big],\ \text{fm}\colon \text{FMi}, \\
& \quad \text{ck}\colon \\
& \quad\quad \text{CK} \\
& \quad\quad\quad \big[\text{Configuration},\ \text{FeatureExpression},\ \text{sat},\ \text{FMi},\ \text{Fea-} \\
& \text{ture},\ [\text{——}],\ \text{wf},\ \text{wt}, \\
& \quad\quad\quad\quad \text{genFeatureExpression},\ \text{getFeatures},\ \text{addMandatory},\ \text{ad-} \\
& \text{dOptional}\big], \\
& \quad \text{anSet})\colon \\
& \quad (s \subseteq (\Diamond)(\text{fm},\ \text{ck},\ \text{anSet})) \Rightarrow (s \subseteq [\text{——}](\text{fm})) \\
\{-2\} & \text{syntaxChangeAsset}(A(\text{pl}),\ \text{am2},\ \text{pairs},\ a_1,\ a_2,\ \text{an}) \\
\{-3\} & s\ =\ (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an})) \\ \hline
\{1\} & (s \subseteq [\text{——}](F(\text{pl})))
\end{array}$$

Instantiating the top quantifier in -1 with the terms: s, F(pl), K(pl), singleton(an),
`changeAssetStrongPartialRef.2`:

$$\begin{array}{|ll}
\{-1\} & (s \subseteq (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an}))) \Rightarrow (s \subseteq [\text{——}](F(\text{pl}))) \\
\{-2\} & \text{syntaxChangeAsset}(A(\text{pl}),\ \text{am2},\ \text{pairs},\ a_1,\ a_2,\ \text{an}) \\
\{-3\} & s\ =\ (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an})) \\ \hline
\{1\} & (s \subseteq [\text{——}](F(\text{pl})))
\end{array}$$

Applying bddsimp,
`changeAssetStrongPartialRef.2`:

$$\begin{array}{|ll}
\{-1\} & \text{syntaxChangeAsset}(A(\text{pl}),\ \text{am2},\ \text{pairs},\ a_1,\ a_2,\ \text{an}) \\
\{-2\} & s\ =\ (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an})) \\ \hline
\{1\} & (s \subseteq (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an}))) \\
\{2\} & (s \subseteq [\text{——}](F(\text{pl})))
\end{array}$$

Replacing using formula -2,
`changeAssetStrongPartialRef.2`:

$$\begin{array}{|ll}
\{-1\} & \text{syntaxChangeAsset}(A(\text{pl}),\ \text{am2},\ \text{pairs},\ a_1,\ a_2,\ \text{an}) \\
\{-2\} & s\ =\ (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an})) \\ \hline
\{1\} & ((\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an})) \subseteq (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an}))) \\
\{2\} & (s \subseteq [\text{——}](F(\text{pl})))
\end{array}$$

Expanding the definition of subset?,

`changeAssetStrongPartialRef.2`:

$$
\begin{array}{|ll}
\{\text{-}1\} & \text{syntaxChangeAsset}(A(\text{pl}),\ \text{am2},\ \text{pairs},\ a_1,\ a_2,\ \text{an}) \\
\{\text{-}2\} & s\ =\ (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an})) \\
\hline
\{1\} & \text{FORALL}\ (x\colon\ \text{Configuration})\colon \\
 & \quad (x \in (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an}))) \Rightarrow \\
 & \quad\ (x \in (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an}))) \\
\{2\} & \text{FORALL}\ (x\colon\ \text{Configuration})\colon\ (x \in s)\ \Rightarrow\ (x \in [\text{——}](F(\text{pl})))
\end{array}
$$

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

$$
\begin{array}{|ll}
\{\text{-}1\} & \text{syntaxChangeAsset}(A(\text{pl}),\ \text{am2},\ \text{pairs},\ a_1,\ a_2,\ \text{an}) \\
\{\text{-}2\} & s\ =\ (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an})) \\
\hline
\{1\} & (c \in (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an}))) \Rightarrow \\
 & \quad (c \in (\Diamond)(F(\text{pl}),\ K(\text{pl}),\ \text{singleton}(\text{an}))) \\
\{2\} & \text{FORALL}\ (x\colon\ \text{Configuration})\colon\ (x \in s)\ \Rightarrow\ (x \in [\text{——}](F(\text{pl})))
\end{array}
$$

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

Verbose proof for `addAssetsStrongPartialRef`.
`addAssetsStrongPartialRef`:

$$
\begin{array}{|l}
\\ \hline
\{1\} \quad \textsc{forall}\ (\mathrm{pl},\ \mathrm{am2},\ \mathrm{ck2},\ s,\ \mathrm{its},\ \mathrm{pairs}):\\
\qquad ((s\ =\ \diamond\ (F(\mathrm{pl2}),\ K(\mathrm{pl2}),\ \mathrm{domain(pairs)})\ \wedge\\
\qquad\qquad \mathrm{syntaxAddAssets}(A(\mathrm{pl}),\ \mathrm{am2},\ K(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{pairs},\ \mathrm{its})\ \wedge\\
\qquad\qquad\ \mathrm{conditionsAddAssets(pairs,\ its)}\ \wedge\\
\qquad\qquad\quad (\textsc{forall}\ c:\\
\qquad\qquad\qquad \neg\ s(c)\ \Rightarrow\\
\qquad\qquad\qquad\ \mathrm{SPLrefinement.wfProduct}(\mathrm{semantics}(K(\mathrm{pl2}))(A(\mathrm{pl2}))(c))))\\
\qquad\quad \Rightarrow\ \mathrm{strongPartialRefinement}(\mathrm{pl},\ \mathrm{pl2},\ s))\\
\qquad\quad \textsc{where}\ \mathrm{pl2}\ =\ (\#F\ :=\ F(\mathrm{pl}),\ A\ :=\ \mathrm{am2},\ K\ :=\ \mathrm{ck2}\#)
\end{array}
$$

`addAssetsStrongPartialRef`:

$$
\begin{array}{|l}
\\ \hline
\{1\} \quad \textsc{forall}\ (\mathrm{pl},\ \mathrm{am2},\ \mathrm{ck2},\ s,\ \mathrm{its},\ \mathrm{pairs}):\\
\qquad ((s\ =\ \diamond\ (F(\mathrm{pl2}),\ K(\mathrm{pl2}),\ \mathrm{domain(pairs)})\ \wedge\\
\qquad\qquad \mathrm{syntaxAddAssets}(A(\mathrm{pl}),\ \mathrm{am2},\ K(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{pairs},\ \mathrm{its})\ \wedge\\
\qquad\qquad\ \mathrm{conditionsAddAssets(pairs,\ its)}\ \wedge\\
\qquad\qquad\quad (\textsc{forall}\ c:\\
\qquad\qquad\qquad \neg\ s(c)\ \Rightarrow\\
\qquad\qquad\qquad\ \mathrm{SPLrefinement.wfProduct}(\mathrm{semantics}(K(\mathrm{pl2}))(A(\mathrm{pl2}))(c))))\\
\qquad\quad \Rightarrow\ \mathrm{strongPartialRefinement}(\mathrm{pl},\ \mathrm{pl2},\ s))\\
\qquad\quad \textsc{where}\ \mathrm{pl2}\ =\ (\#F\ :=\ F(\mathrm{pl}),\ A\ :=\ \mathrm{am2},\ K\ :=\ \mathrm{ck2}\#)
\end{array}
$$

For the top quantifier in 1, we introduce Skolem constants: (pl am2 ck2 s its pairs),
`addAssetsStrongPartialRef`:

$$
\begin{array}{|l}
\\ \hline
\{1\} \quad ((s\ =\ (\diamond)(F(\mathrm{pl2}),\ K(\mathrm{pl2}),\ \mathrm{domain(pairs)})\ \wedge\\
\qquad\qquad \mathrm{syntaxAddAssets}(A(\mathrm{pl}),\ \mathrm{am2},\ K(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{pairs},\ \mathrm{its})\ \wedge\\
\qquad\qquad\ \mathrm{conditionsAddAssets(pairs,\ its)}\ \wedge\\
\qquad\qquad\quad (\textsc{forall}\ c:\\
\qquad\qquad\qquad \neg\ s(c)\ \Rightarrow\ \mathrm{SPLrefinement.wfProduct}(\mathrm{semantics}(K(\mathrm{pl2}))(A(\mathrm{pl2}))(c))))\\
\qquad\quad \Rightarrow\ \mathrm{strongPartialRefinement}(\mathrm{pl},\ \mathrm{pl2},\ s))\\
\qquad\quad \textsc{where}\ \mathrm{pl2}\ =\ (\#F\ :=\ F(\mathrm{pl}),\ A\ :=\ \mathrm{am2},\ K\ :=\ \mathrm{ck2}\#)
\end{array}
$$

Expanding the definition of strongPartialRefinement, `addAssetsStrongPartialRef`:

$$
\begin{array}{l|ll}
\hline
\{1\} & ((s = (\Diamond)(F(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{domain}(\mathrm{pairs})) \wedge \\
& \quad \mathrm{syntaxAddAssets}(A(\mathrm{pl}),\ \mathrm{am2},\ K(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{pairs},\ \mathrm{its}) \wedge \\
& \quad \mathrm{conditionsAddAssets}(\mathrm{pairs},\ \mathrm{its}) \wedge \\
& \quad (\textsc{forall}\ c\colon \neg\ s(c) \Rightarrow \mathrm{SPLrefinement.wfProduct}(\mathrm{semantics}(\mathrm{ck2})(\mathrm{am2})(c)))) \\
& \Rightarrow \\
& (s \subseteq [\text{——}](F(\mathrm{pl}))) \wedge \\
& (s \subseteq [\text{——}](F(\mathrm{pl}))) \wedge \\
& (\textsc{forall}\ (c\colon \mathrm{Configuration})\colon \\
& \quad s(c) \Rightarrow (\mathrm{prod}(\mathrm{pl},\ c)\ \text{—-}\ \mathrm{prod}((\#F := F(\mathrm{pl}),\ A := \mathrm{am2},\ K := \mathrm{ck2}\#), \\
\end{array}
$$

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

$$
\begin{array}{l|ll}
\{-1\} & s = (\Diamond)(F(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{domain}(\mathrm{pairs})) \\
\{-2\} & \mathrm{syntaxAddAssets}(A(\mathrm{pl}),\ \mathrm{am2},\ K(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{pairs},\ \mathrm{its}) \\
\{-3\} & \mathrm{conditionsAddAssets}(\mathrm{pairs},\ \mathrm{its}) \\
\{-4\} & \textsc{forall}\ c\colon \neg\ s(c) \Rightarrow \mathrm{SPLrefinement.wfProduct}(\mathrm{semantics}(\mathrm{ck2})(\mathrm{am2})(c)) \\
\hline
\{1\} & \textsc{forall}\ (c\colon \mathrm{Configuration})\colon \\
& \quad s(c) \Rightarrow (\mathrm{prod}(\mathrm{pl},\ c)\ \text{—-}\ \mathrm{prod}((\#F := F(\mathrm{pl}),\ A := \mathrm{am2},\ K := \mathrm{ck2}\#),\ c)) \\
\end{array}
$$

Applying addAssetsSameProducts

addAssetsStrongPartialRef.1:

{-1} ∀ (fm: FMi, am,
ck:
CK
[Configuration, FeatureExpression, sat, FMi, Feature, [——], wf, wt,
genFeatureExpression, getFeatures, addMandatory, addOptional],
am2,
ck2:
CK
[Configuration, FeatureExpression, sat, FMi, Feature, [——], wf, wt,
genFeatureExpression, getFeatures, addMandatory, addOptional],
s: set[Configuration],
its:
set
[Item
[Configuration, FeatureExpression, sat, FMi, Feature, [——], wf, wt,
genFeatureExpression, getFeatures, addMandatory, addOptional]],
pairs):
((s = (◇)(fm, ck2, domain(pairs)) ∧
syntaxAddAssets(am, am2, ck, ck2, pairs, its) ∧ conditionsAddAssets(pairs, its))
⇒
(FORALL (c: Configuration):
$s(c)$ ⇒ ((semantics(ck)(am)(c)) = (semantics(ck2)(am2)(c)))))

{-2} s = (◇)(F(pl), ck2, domain(pairs))

{-3} syntaxAddAssets(A(pl), am2, K(pl), ck2, pairs, its)

{-4} conditionsAddAssets(pairs, its)

{-5} FORALL c: ¬ $s(c)$ ⇒ SPLrefinement.wfProduct(semantics(ck2)(am2)(c))

{1} FORALL (c: Configuration):
$s(c)$ ⇒ (prod(pl, c) ⊑ prod((#F := F(pl), A := am2, K := ck2#), c))

Instantiating the top quantifier in -1 with the terms: F(pl), A(pl), K(pl), am2, ck2, s, its, pairs,

`addAssetsStrongPartialRef.1:`

```
{-1}  ((s = (◇)(F(pl), ck2, domain(pairs)) ∧
             syntaxAddAssets(A(pl), am2, K(pl), ck2, pairs, its) ∧
              conditionsAddAssets(pairs, its))
           ⇒
           (FORALL (c: Configuration):
                s(c) ⇒
                 ((semantics(K(pl))(A(pl))(c)) = (semantics(ck2)(am2)(c)))))
{-2}  s = (◇)(F(pl), ck2, domain(pairs))
{-3}  syntaxAddAssets(A(pl), am2, K(pl), ck2, pairs, its)
{-4}  conditionsAddAssets(pairs, its)
{-5}  FORALL c: ¬ s(c) ⇒ SPLrefinement.wfProduct(semantics(ck2)(am2)(c))
---------------------------------------------------------------------------
 {1}  FORALL (c: Configuration):
        s(c) ⇒ (prod(pl, c) —- prod((#F := F(pl), A := am2, K := ck2#), c))
```

Applying bddsimp,

`addAssetsStrongPartialRef.1:`

```
{-1}  s = (◇)(F(pl), ck2, domain(pairs))
{-2}  syntaxAddAssets(A(pl), am2, K(pl), ck2, pairs, its)
{-3}  conditionsAddAssets(pairs, its)
{-4}  FORALL (c: Configuration):
        s(c) ⇒ ((semantics(K(pl))(A(pl))(c)) = (semantics(ck2)(am2)(c)))
{-5}  FORALL c: ¬ s(c) ⇒ SPLrefinement.wfProduct(semantics(ck2)(am2)(c))
---------------------------------------------------------------------------
 {1}  FORALL (c: Configuration):
        s(c) ⇒ (prod(pl, c) —- prod((#F := F(pl), A := am2, K := ck2#), c))
```

Expanding the definition of prod,

`addAssetsStrongPartialRef.1:`

```
{-1}  s = (◇)(F(pl), ck2, domain(pairs))
{-2}  syntaxAddAssets(A(pl), am2, K(pl), ck2, pairs, its)
{-3}  conditionsAddAssets(pairs, its)
{-4}  FORALL (c: Configuration):
        s(c) ⇒ ((semantics(K(pl))(A(pl))(c)) = (semantics(ck2)(am2)(c)))
{-5}  FORALL c: ¬ s(c) ⇒ SPLrefinement.wfProduct(semantics(ck2)(am2)(c))
---------------------------------------------------------------------------
 {1}  FORALL (c: Configuration):
        s(c) ⇒ ((semantics(K(pl))(A(pl))(c)) —- (semantics(ck2)(am2)(c)))
```

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

`addAssetsStrongPartialRef.1:`

{-1} s = ($\Diamond$)(F(pl), ck2, domain(pairs))
{-2} syntaxAddAssets(A(pl), am2, K(pl), ck2, pairs, its)
{-3} conditionsAddAssets(pairs, its)
{-4} FORALL (c: Configuration):
    $s(c)$ $\Rightarrow$ ((semantics(K(pl))(A(pl))(c)) = (semantics(ck2)(am2)(c)))
{-5} FORALL c: $\neg$ $s(c)$ $\Rightarrow$ SPLrefinement.wfProduct(semantics(ck2)(am2)(c))

{1} $s(c)$ $\Rightarrow$ ((semantics(K(pl))(A(pl))(c)) —- (semantics(ck2)(am2)(c)))

Instantiating the top quantifier in -4 with the terms: c,
`addAssetsStrongPartialRef.1:`

{-1} s = ($\Diamond$)(F(pl), ck2, domain(pairs))
{-2} syntaxAddAssets(A(pl), am2, K(pl), ck2, pairs, its)
{-3} conditionsAddAssets(pairs, its)
{-4} $s(c)$ $\Rightarrow$ ((semantics(K(pl))(A(pl))(c)) = (semantics(ck2)(am2)(c)))
{-5} FORALL c: $\neg$ $s(c)$ $\Rightarrow$ SPLrefinement.wfProduct(semantics(ck2)(am2)(c))

{1} $s(c)$ $\Rightarrow$ ((semantics(K(pl))(A(pl))(c)) —- (semantics(ck2)(am2)(c)))

Applying bddsimp,
`addAssetsStrongPartialRef.1:`

{-1} s = ($\Diamond$)(F(pl), ck2, domain(pairs))
{-2} syntaxAddAssets(A(pl), am2, K(pl), ck2, pairs, its)
{-3} conditionsAddAssets(pairs, its)
{-4} $s(c)$
{-5} ((semantics(K(pl))(A(pl))(c)) = (semantics(ck2)(am2)(c)))
{-6} FORALL c: $\neg$ $s(c)$ $\Rightarrow$ SPLrefinement.wfProduct(semantics(ck2)(am2)(c))

{1} ((semantics(K(pl))(A(pl))(c)) —- (semantics(ck2)(am2)(c)))

Replacing using formula -5,
`addAssetsStrongPartialRef.1:`

{-1} s = ($\Diamond$)(F(pl), ck2, domain(pairs))
{-2} syntaxAddAssets(A(pl), am2, K(pl), ck2, pairs, its)
{-3} conditionsAddAssets(pairs, its)
{-4} $s(c)$
{-5} ((semantics(K(pl))(A(pl))(c)) = (semantics(ck2)(am2)(c)))
{-6} FORALL c: $\neg$ $s(c)$ $\Rightarrow$ SPLrefinement.wfProduct(semantics(ck2)(am2)(c))

{1} ((semantics(ck2)(am2)(c)) —- (semantics(ck2)(am2)(c)))

Using lemma SPLrefinement.assetRefinement, addAssetsStrongPartialRef.1:

{-1} orders[set[Asset]].preorder?(—-)
{-2} s = (◇)(F(pl), ck2, domain(pairs))
{-3} syntaxAddAssets(A(pl), am2, K(pl), ck2, pairs, its)
{-4} conditionsAddAssets(pairs, its)
{-5} $s(c)$
{-6} ((semantics(K(pl))(A(pl))(c)) = (semantics(ck2)(am2)(c)))
{-7} FORALL c: ¬ $s(c)$ ⇒ SPLrefinement.wfProduct(semantics(ck2)(am2)(c))

{1} ((semantics(ck2)(am2)(c)) —- (semantics(ck2)(am2)(c)))

Expanding the definition of preorder?,
addAssetsStrongPartialRef.1:

{-1} reflexive?(—-) & transitive?(—-)
{-2} s = (◇)(F(pl), ck2, domain(pairs))
{-3} syntaxAddAssets(A(pl), am2, K(pl), ck2, pairs, its)
{-4} conditionsAddAssets(pairs, its)
{-5} $s(c)$
{-6} ((semantics(K(pl))(A(pl))(c)) = (semantics(ck2)(am2)(c)))
{-7} FORALL c: ¬ $s(c)$ ⇒ SPLrefinement.wfProduct(semantics(ck2)(am2)(c))

{1} ((semantics(ck2)(am2)(c)) —- (semantics(ck2)(am2)(c)))

Applying disjunctive simplification to flatten sequent,
addAssetsStrongPartialRef.1:

{-1} reflexive?(—-)
{-2} transitive?(—-)
{-3} s = (◇)(F(pl), ck2, domain(pairs))
{-4} syntaxAddAssets(A(pl), am2, K(pl), ck2, pairs, its)
{-5} conditionsAddAssets(pairs, its)
{-6} $s(c)$
{-7} ((semantics(K(pl))(A(pl))(c)) = (semantics(ck2)(am2)(c)))
{-8} FORALL c: ¬ $s(c)$ ⇒ SPLrefinement.wfProduct(semantics(ck2)(am2)(c))

{1} ((semantics(ck2)(am2)(c)) —- (semantics(ck2)(am2)(c)))

Expanding the definition of reflexive?,

`addAssetsStrongPartialRef.1`:

{-1}	FORALL $(x\colon \text{set}[\text{Asset}])\colon (x \longrightarrow x)$
{-2}	transitive?(—-)
{-3}	$s = (\Diamond)(F(\text{pl}),\ \text{ck2},\ \text{domain(pairs)})$
{-4}	syntaxAddAssets(A(pl), am2, K(pl), ck2, pairs, its)
{-5}	conditionsAddAssets(pairs, its)
{-6}	$s(c)$
{-7}	((semantics(K(pl))(A(pl))(c)) = (semantics(ck2)(am2)(c)))
{-8}	FORALL c: $\neg$ $s(c)$ $\Rightarrow$ SPLrefinement.wfProduct(semantics(ck2)(am2)(c))
{1}	((semantics(ck2)(am2)(c)) —- (semantics(ck2)(am2)(c)))

Instantiating the top quantifier in -1 with the terms: (semantics(ck2)(am2)(c)),
This completes the proof of `addAssetsStrongPartialRef.1`.
`addAssetsStrongPartialRef.2`:

{-1}	$s = (\Diamond)(F(\text{pl}),\ \text{ck2},\ \text{domain(pairs)})$
{-2}	syntaxAddAssets(A(pl), am2, K(pl), ck2, pairs, its)
{-3}	conditionsAddAssets(pairs, its)
{-4}	FORALL c: $\neg$ $s(c)$ $\Rightarrow$ SPLrefinement.wfProduct(semantics(ck2)(am2)(c))
{1}	$(s \subseteq [\text{——}](F(\text{pl})))$

Applying filteredConfigurations
`addAssetsStrongPartialRef.2`:

{-1}	∀ (s: set[Configuration], fm: FMi, ck: CK [Configuration, FeatureExpression, sat, FMi, Feature, [——], wf, wt, genFeatureExpression, getFeatures, addMandatory, addOptional], anSet): $(s \subseteq (\Diamond)(\text{fm},\ \text{ck},\ \text{anSet})) \Rightarrow (s \subseteq [\text{——}](\text{fm}))$
{-2}	$s = (\Diamond)(F(\text{pl}),\ \text{ck2},\ \text{domain(pairs)})$
{-3}	syntaxAddAssets(A(pl), am2, K(pl), ck2, pairs, its)
{-4}	conditionsAddAssets(pairs, its)
{-5}	FORALL c: $\neg$ $s(c)$ $\Rightarrow$ SPLrefinement.wfProduct(semantics(ck2)(am2)(c))
{1}	$(s \subseteq [\text{——}](F(\text{pl})))$

Instantiating the top quantifier in -1 with the terms: s, F(pl), ck2, domain(pairs),

`addAssetsStrongPartialRef.2:`

$$
\begin{array}{|ll}
\{\text{-}1\} & (s \subseteq (\Diamond)(F(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{domain}(\mathrm{pairs}))) \Rightarrow (s \subseteq [\text{——}](F(\mathrm{pl}))) \\
\{\text{-}2\} & s = (\Diamond)(F(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{domain}(\mathrm{pairs})) \\
\{\text{-}3\} & \mathrm{syntaxAddAssets}(A(\mathrm{pl}),\ \mathrm{am2},\ K(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{pairs},\ \mathrm{its}) \\
\{\text{-}4\} & \mathrm{conditionsAddAssets}(\mathrm{pairs},\ \mathrm{its}) \\
\{\text{-}5\} & \textsc{forall}\ c{:}\ \neg\ s(c) \Rightarrow \mathrm{SPLrefinement.wfProduct}(\mathrm{semantics}(\mathrm{ck2})(\mathrm{am2})(c)) \\
\hline
\{1\} & (s \subseteq [\text{——}](F(\mathrm{pl})))
\end{array}
$$

Applying bddsimp,
`addAssetsStrongPartialRef.2:`

$$
\begin{array}{|ll}
\{\text{-}1\} & s = (\Diamond)(F(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{domain}(\mathrm{pairs})) \\
\{\text{-}2\} & \mathrm{syntaxAddAssets}(A(\mathrm{pl}),\ \mathrm{am2},\ K(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{pairs},\ \mathrm{its}) \\
\{\text{-}3\} & \mathrm{conditionsAddAssets}(\mathrm{pairs},\ \mathrm{its}) \\
\{\text{-}4\} & \textsc{forall}\ c{:}\ \neg\ s(c) \Rightarrow \mathrm{SPLrefinement.wfProduct}(\mathrm{semantics}(\mathrm{ck2})(\mathrm{am2})(c)) \\
\hline
\{1\} & (s \subseteq (\Diamond)(F(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{domain}(\mathrm{pairs}))) \\
\{2\} & (s \subseteq [\text{——}](F(\mathrm{pl})))
\end{array}
$$

Replacing using formula -1,
`addAssetsStrongPartialRef.2:`

$$
\begin{array}{|ll}
\{\text{-}1\} & s = (\Diamond)(F(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{domain}(\mathrm{pairs})) \\
\{\text{-}2\} & \mathrm{syntaxAddAssets}(A(\mathrm{pl}),\ \mathrm{am2},\ K(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{pairs},\ \mathrm{its}) \\
\{\text{-}3\} & \mathrm{conditionsAddAssets}(\mathrm{pairs},\ \mathrm{its}) \\
\{\text{-}4\} & \textsc{forall}\ c{:}\ \neg\ s(c) \Rightarrow \mathrm{SPLrefinement.wfProduct}(\mathrm{semantics}(\mathrm{ck2})(\mathrm{am2})(c)) \\
\hline
\{1\} & ((\Diamond)(F(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{domain}(\mathrm{pairs})) \subseteq (\Diamond)(F(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{domain}(\mathrm{pairs}))) \\
\{2\} & (s \subseteq [\text{——}](F(\mathrm{pl})))
\end{array}
$$

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

$$
\begin{array}{|ll}
\{\text{-}1\} & s = (\Diamond)(F(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{domain}(\mathrm{pairs})) \\
\{\text{-}2\} & \mathrm{syntaxAddAssets}(A(\mathrm{pl}),\ \mathrm{am2},\ K(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{pairs},\ \mathrm{its}) \\
\{\text{-}3\} & \mathrm{conditionsAddAssets}(\mathrm{pairs},\ \mathrm{its}) \\
\{\text{-}4\} & \textsc{forall}\ c{:}\ \neg\ s(c) \Rightarrow \mathrm{SPLrefinement.wfProduct}(\mathrm{semantics}(\mathrm{ck2})(\mathrm{am2})(c)) \\
\hline
\{1\} & \textsc{forall}\ (x{:}\ \mathrm{Configuration}){:} \\
 & \quad (x \in (\Diamond)(F(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{domain}(\mathrm{pairs}))) \Rightarrow \\
 & \quad\ (x \in (\Diamond)(F(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{domain}(\mathrm{pairs}))) \\
\{2\} & \textsc{forall}\ (x{:}\ \mathrm{Configuration}){:}\ (x \in s) \Rightarrow (x \in [\text{——}](F(\mathrm{pl})))
\end{array}
$$

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

`addAssetsStrongPartialRef.2`:

$$
\begin{array}{|ll}
\{\text{-}1\} & s \ = \ (\Diamond)(F(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{domain}(\mathrm{pairs})) \\
\{\text{-}2\} & \mathrm{syntaxAddAssets}(A(\mathrm{pl}),\ \mathrm{am2},\ K(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{pairs},\ \mathrm{its}) \\
\{\text{-}3\} & \mathrm{conditionsAddAssets}(\mathrm{pairs},\ \mathrm{its}) \\
\{\text{-}4\} & \textsc{forall}\ c\colon\ \neg\ s(c)\ \Rightarrow\ \mathrm{SPLrefinement.wfProduct}(\mathrm{semantics}(\mathrm{ck2})(\mathrm{am2})(c)) \\
\hline
\{1\} & (c \in (\Diamond)(F(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{domain}(\mathrm{pairs})))\ \Rightarrow \\
 & \quad (c \in (\Diamond)(F(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{domain}(\mathrm{pairs}))) \\
\{2\} & \textsc{forall}\ (x\colon\ \mathrm{Configuration})\colon\ (x \in s)\ \Rightarrow\ (x \in [\text{——}](F(\mathrm{pl})))
\end{array}
$$

Applying bddsimp,

This completes the proof of `addAssetsStrongPartialRef.2`.

Q.E.D.

Verbose proof for `changeCKLineStrongPartialRef`.
`changeCKLineStrongPartialRef`:

$$
\begin{array}{l}
\{1\}\quad \text{FORALL } (\mathrm{pl},\ \mathrm{ck2},\ \mathrm{item1},\ \mathrm{item2},\ \mathrm{its},\ s):\\
\quad ((\mathrm{wfCK}(F(\mathrm{pl}),\ A(\mathrm{pl}),\ K(\mathrm{pl}))\ \wedge\\
\quad\quad s\ =\ (\Diamond\ (F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item1})) \cap \Diamond\ (F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2})))\ \wedge\\
\quad\quad \mathrm{syntaxChangeCKLine}(K(\mathrm{pl}),\ K(\mathrm{pl2}),\ \mathrm{item1},\ \mathrm{item2},\ \mathrm{its})\ \wedge\\
\quad\quad \mathrm{wt}(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2})))\\
\quad\quad \Rightarrow\ \mathrm{strongPartialRefinement}(\mathrm{pl},\ \mathrm{pl2},\ s))\\
\quad\quad \text{WHERE } \mathrm{pl2}\ =\ (\#F\ :=\ F(\mathrm{pl}),\ A\ :=\ A(\mathrm{pl}),\ K\ :=\ \mathrm{ck2}\#)
\end{array}
$$

For the top quantifier in 1, we introduce Skolem constants: (pl ck2 item1 item2 its s),
`changeCKLineStrongPartialRef`:

$$
\begin{array}{l}
\{1\}\quad ((\mathrm{wfCK}(F(\mathrm{pl}),\ A(\mathrm{pl}),\ K(\mathrm{pl}))\ \wedge\\
\quad\quad s\ =\ ((\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item1})) \cap (\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2})))\ \wedge\\
\quad\quad \mathrm{syntaxChangeCKLine}(K(\mathrm{pl}),\ K(\mathrm{pl2}),\ \mathrm{item1},\ \mathrm{item2},\ \mathrm{its})\ \wedge\\
\quad\quad \mathrm{wt}(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2})))\\
\quad\quad \Rightarrow\ \mathrm{strongPartialRefinement}(\mathrm{pl},\ \mathrm{pl2},\ s))\\
\quad\quad \text{WHERE } \mathrm{pl2}\ =\ (\#F\ :=\ F(\mathrm{pl}),\ A\ :=\ A(\mathrm{pl}),\ K\ :=\ \mathrm{ck2}\#)
\end{array}
$$

Expanding the definition of strongPartialRefinement,
`changeCKLineStrongPartialRef`:

$$
\begin{array}{l}
\{1\}\quad ((\mathrm{wfCK}(F(\mathrm{pl}),\ A(\mathrm{pl}),\ K(\mathrm{pl}))\ \wedge\\
\quad\quad s\ =\ ((\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item1})) \cap (\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2})))\ \wedge\\
\quad\quad \mathrm{syntaxChangeCKLine}(K(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{item1},\ \mathrm{item2},\ \mathrm{its})\ \wedge\ \mathrm{wt}(F(\mathrm{pl}),\ \text{ge-}\\
\quad \text{tExp}(\mathrm{item2})))\\
\quad\quad \Rightarrow\\
\quad\quad (s \subseteq [\text{——}](F(\mathrm{pl})))\ \wedge\\
\quad\quad (s \subseteq [\text{——}](F(\mathrm{pl})))\ \wedge\\
\quad\quad (\text{FORALL } (c\colon\ \mathrm{Configuration}):\\
\quad\quad\quad s(c)\ \Rightarrow\\
\quad\quad\quad (\mathrm{prod}(\mathrm{pl},\ c)\ \text{—-}\ \mathrm{prod}((\#F\ :=\ F(\mathrm{pl}),\ A\ :=\ A(\mathrm{pl}),\ K\ :=\ \mathrm{ck2}\#),\ c))))
\end{array}
$$

Applying bddsimp,
we get 2 subgoals:

`changeCKLineStrongPartialRef.1:`

$$
\begin{array}{l|ll}
 & \{-1\} & \mathrm{wfCK}(F(\mathrm{pl}),\ A(\mathrm{pl}),\ K(\mathrm{pl})) \\
 & \{-2\} & s\ =\ ((\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item1})) \cap (\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2}))) \\
 & \{-3\} & \mathrm{syntaxChangeCKLine}(K(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{item1},\ \mathrm{item2},\ \mathrm{its}) \\
 & \{-4\} & \mathrm{wt}(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2})) \\
\hline
 & \{1\} & \textsc{forall}\ (c\colon \mathrm{Configuration})\colon \\
 & & \quad s(c)\ \Rightarrow\ (\mathrm{prod}(\mathrm{pl},\ c)\ \text{—-}\ \mathrm{prod}((\#F\ :=\ F(\mathrm{pl}),\ A\ :=\ A(\mathrm{pl}),\ K\ :=\ \mathrm{ck2}\#),\ c))
\end{array}
$$

Applying falseExpMakesNoDiff
`changeCKLineStrongPartialRef.1:`

$$
\begin{array}{l|ll}
 & \{-1\} & \forall\ (\mathrm{fm}\colon \mathrm{FMi},\ \mathrm{am}, \\
 & & \qquad \mathrm{ck1}\colon \\
 & & \qquad\quad \mathrm{CK} \\
 & & \qquad\qquad [\mathrm{Configuration},\ \mathrm{FeatureExpression},\ \mathrm{sat},\ \mathrm{FMi},\ \mathrm{Fea\text{-}} \\
 & & \mathrm{ture},\ [\text{——}],\ \mathrm{wf},\ \mathrm{wt}, \\
 & & \qquad\qquad\quad \mathrm{genFeatureExpression},\ \mathrm{getFeatures},\ \mathrm{addMandatory},\ \mathrm{ad\text{-}} \\
 & & \mathrm{dOptional}], \\
 & & \qquad \mathrm{ck2}\colon \\
 & & \qquad\quad \mathrm{CK} \\
 & & \qquad\qquad [\mathrm{Configuration},\ \mathrm{FeatureExpression},\ \mathrm{sat},\ \mathrm{FMi},\ \mathrm{Fea\text{-}} \\
 & & \mathrm{ture},\ [\text{——}],\ \mathrm{wf},\ \mathrm{wt}, \\
 & & \qquad\qquad\quad \mathrm{genFeatureExpression},\ \mathrm{getFeatures},\ \mathrm{addMandatory},\ \mathrm{ad\text{-}} \\
 & & \mathrm{dOptional}], \\
 & & \qquad s\colon\ \mathrm{set}[\mathrm{Configuration}])\colon \\
 & & \quad \textsc{forall}\ (c\colon \mathrm{Configuration})\colon \\
 & & \quad\quad s(c)\ \Rightarrow \\
 & & \qquad ((\textsc{forall}\ (\mathrm{item}\colon \\
 & & \qquad\qquad\qquad \mathrm{Item} \\
 & & \qquad\qquad\qquad\quad [\mathrm{Configuration},\ \mathrm{FeatureExpres\text{-}} \\
 & & \mathrm{sion},\ \mathrm{sat},\ \mathrm{FMi},\ \mathrm{Feature},\ [\text{——}],\ \mathrm{wf},\ \mathrm{wt}, \\
 & & \qquad\qquad\qquad\qquad \mathrm{genFeatureExpression},\ \mathrm{getFeatures},\ \mathrm{ad\text{-}} \\
 & & \mathrm{dMandatory},\ \mathrm{addOptional}])\colon \\
 & & \qquad\qquad \mathrm{diffIts}(\mathrm{item})\ \Rightarrow\ \neg\ \mathrm{sat}(\mathrm{getExp}(\mathrm{item}),\ c)) \\
 & & \qquad\quad \Rightarrow\ (\mathrm{semantics}(\mathrm{ck1})(\mathrm{am})(c)\ =\ \mathrm{semantics}(\mathrm{ck2})(\mathrm{am})(c))) \\
 & & \quad\quad \textsc{where}\ \mathrm{diffIts}\ =\ \mathrm{symmetric\_difference}(\mathrm{items}(\mathrm{ck1}),\ \mathrm{items}(\mathrm{ck2})) \\
 & \{-2\} & \mathrm{wfCK}(F(\mathrm{pl}),\ A(\mathrm{pl}),\ K(\mathrm{pl})) \\
 & \{-3\} & s\ =\ ((\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item1})) \cap (\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2}))) \\
 & \{-4\} & \mathrm{syntaxChangeCKLine}(K(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{item1},\ \mathrm{item2},\ \mathrm{its}) \\
 & \{-5\} & \mathrm{wt}(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2})) \\
\hline
 & \{1\} & \textsc{forall}\ (c\colon \mathrm{Configuration})\colon \\
 & & \quad s(c)\ \Rightarrow\ (\mathrm{prod}(\mathrm{pl},\ c)\ \text{—-}\ \mathrm{prod}((\#F\ :=\ F(\mathrm{pl}),\ A\ :=\ A(\mathrm{pl}),\ K\ :=\ \mathrm{ck2}\#),\ c))
\end{array}
$$

Instantiating the top quantifier in -1 with the terms: $F(\mathrm{pl})$, $A(\mathrm{pl})$, $K(\mathrm{pl})$, ck2, s,

`changeCKLineStrongPartialRef.1:`

$$
\begin{array}{l|ll}
 & \{-1\} & \textsc{forall}\ (c\colon \text{Configuration}): \\
 & & \quad s(c) \Rightarrow \\
 & & \quad\ ((\textsc{forall}\ (\text{item}: \\
 & & \qquad\qquad\qquad \text{Item} \\
 & & \qquad\qquad\qquad\quad [\text{Configuration},\ \text{FeatureExpres-} \\
 & & \text{sion},\ \text{sat},\ \text{FMi},\ \text{Feature},\ [\text{——}],\ \text{wf},\ \text{wt}, \\
 & & \qquad\qquad\qquad\quad \text{genFeatureExpression},\ \text{getFeatures},\ \text{ad-} \\
 & & \text{dMandatory},\ \text{addOptional}]): \\
 & & \qquad\quad \text{diffIts}(\text{item}) \Rightarrow \neg\ \text{sat}(\text{getExp}(\text{item}),\ c)) \\
 & & \qquad\ \Rightarrow (\text{semantics}(K(\text{pl}))(A(\text{pl}))(c) = \text{semantics}(\text{ck2})(A(\text{pl}))(c))) \\
 & & \quad \textsc{where}\ \text{diffIts} = \text{symmetric\_difference}(\text{items}(K(\text{pl})),\ \text{items}(\text{ck2})) \\
 & \{-2\} & \text{wfCK}(F(\text{pl}),\ A(\text{pl}),\ K(\text{pl})) \\
 & \{-3\} & s = ((\Diamond)(F(\text{pl}),\ \text{getExp}(\text{item1})) \cap (\Diamond)(F(\text{pl}),\ \text{getExp}(\text{item2}))) \\
 & \{-4\} & \text{syntaxChangeCKLine}(K(\text{pl}),\ \text{ck2},\ \text{item1},\ \text{item2},\ \text{its}) \\
 & \{-5\} & \text{wt}(F(\text{pl}),\ \text{getExp}(\text{item2})) \\
\hline
 & \{1\} & \textsc{forall}\ (c\colon \text{Configuration}): \\
 & & \quad s(c) \Rightarrow (\text{prod}(\text{pl},\ c)\ \text{—-}\ \text{prod}((\#F := F(\text{pl}),\ A := A(\text{pl}),\ K := \text{ck2}\#),\ c))
\end{array}
$$

Expanding the definition of symmetric_difference,
`changeCKLineStrongPartialRef.1:`

$$
\begin{array}{l|ll}
 & \{-1\} & \textsc{forall}\ (c\colon \text{Configuration}): \\
 & & \quad s(c) \Rightarrow \\
 & & \quad\ ((\textsc{forall}\ (\text{item}: \\
 & & \qquad\qquad\qquad \text{Item} \\
 & & \qquad\qquad\qquad\quad [\text{Configuration},\ \text{FeatureExpres-} \\
 & & \text{sion},\ \text{sat},\ \text{FMi},\ \text{Feature},\ [\text{——}],\ \text{wf},\ \text{wt}, \\
 & & \qquad\qquad\qquad\quad \text{genFeatureExpression},\ \text{getFeatures},\ \text{ad-} \\
 & & \text{dMandatory},\ \text{addOptional}]): \\
 & & \qquad\quad \text{union}((\text{items}(K(\text{pl})) \setminus \text{items}(\text{ck2})),\ (\text{items}(\text{ck2}) \setminus \text{items}(K(\text{pl})))) \\
 & & \qquad\qquad (\text{item}) \\
 & & \qquad\quad\ \Rightarrow \neg\ \text{sat}(\text{getExp}(\text{item}),\ c)) \\
 & & \qquad\ \Rightarrow (\text{semantics}(K(\text{pl}))(A(\text{pl}))(c) = \text{semantics}(\text{ck2})(A(\text{pl}))(c))) \\
 & \{-2\} & \text{wfCK}(F(\text{pl}),\ A(\text{pl}),\ K(\text{pl})) \\
 & \{-3\} & s = ((\Diamond)(F(\text{pl}),\ \text{getExp}(\text{item1})) \cap (\Diamond)(F(\text{pl}),\ \text{getExp}(\text{item2}))) \\
 & \{-4\} & \text{syntaxChangeCKLine}(K(\text{pl}),\ \text{ck2},\ \text{item1},\ \text{item2},\ \text{its}) \\
 & \{-5\} & \text{wt}(F(\text{pl}),\ \text{getExp}(\text{item2})) \\
\hline
 & \{1\} & \textsc{forall}\ (c\colon \text{Configuration}): \\
 & & \quad s(c) \Rightarrow (\text{prod}(\text{pl},\ c)\ \text{—-}\ \text{prod}((\#F := F(\text{pl}),\ A := A(\text{pl}),\ K := \text{ck2}\#),\ c))
\end{array}
$$

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

`changeCKLineStrongPartialRef.1:`

$$
\begin{array}{l|ll}
& \{-1\} & \text{FORALL } (c\colon \text{Configuration}): \\
& & \quad s(c) \Rightarrow \\
& & \quad\; ((\text{FORALL } (\text{item}: \\
& & \qquad\qquad \text{Item} \\
& & \qquad\qquad\quad [\text{Configuration}, \text{FeatureExpres-} \\
& & \text{sion}, \text{sat}, \text{FMi}, \text{Feature}, [\text{——}], \text{wf}, \text{wt}, \\
& & \qquad\qquad\quad \text{genFeatureExpression}, \text{getFeatures}, \text{ad-} \\
& & \text{dMandatory}, \text{addOptional}]): \\
& & \qquad \text{union}((\text{items}(K(\text{pl})) \setminus \text{items}(\text{ck2})), (\text{items}(\text{ck2}) \setminus \text{items}(K(\text{pl})))) \\
& & \qquad\quad (\text{item}) \\
& & \qquad\; \Rightarrow \neg\ \text{sat}(\text{getExp}(\text{item}), c)) \\
& & \quad \Rightarrow (\text{semantics}(K(\text{pl}))(A(\text{pl}))(c) = \text{semantics}(\text{ck2})(A(\text{pl}))(c))) \\
& \{-2\} & \text{wfCK}(F(\text{pl}), A(\text{pl}), K(\text{pl})) \\
& \{-3\} & s = ((\Diamond)(F(\text{pl}), \text{getExp}(\text{item1})) \cap (\Diamond)(F(\text{pl}), \text{getExp}(\text{item2}))) \\
& \{-4\} & \text{syntaxChangeCKLine}(K(\text{pl}), \text{ck2}, \text{item1}, \text{item2}, \text{its}) \\
& \{-5\} & \text{wt}(F(\text{pl}), \text{getExp}(\text{item2})) \\
\hline
& \{1\} & s(c) \Rightarrow (\text{prod}(\text{pl}, c) \text{ —- } \text{prod}((\#F := F(\text{pl}), A := A(\text{pl}), K := \text{ck2}\#), c))
\end{array}
$$

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

$$
\begin{array}{l|ll}
& \{-1\} & s(c) \Rightarrow \\
& & \quad ((\text{FORALL } (\text{item}: \\
& & \qquad\qquad \text{Item} \\
& & \qquad\qquad\quad [\text{Configuration}, \text{FeatureExpres-} \\
& & \text{sion}, \text{sat}, \text{FMi}, \text{Feature}, [\text{——}], \text{wf}, \text{wt}, \\
& & \qquad\qquad\quad \text{genFeatureExpression}, \text{getFeatures}, \text{addManda-} \\
& & \text{tory}, \text{addOptional}]): \\
& & \qquad \text{union}((\text{items}(K(\text{pl})) \setminus \text{items}(\text{ck2})), (\text{items}(\text{ck2}) \setminus \text{items}(K(\text{pl})))) \\
& & \qquad\quad (\text{item}) \\
& & \qquad\; \Rightarrow \neg\ \text{sat}(\text{getExp}(\text{item}), c)) \\
& & \quad \Rightarrow (\text{semantics}(K(\text{pl}))(A(\text{pl}))(c) = \text{semantics}(\text{ck2})(A(\text{pl}))(c))) \\
& \{-2\} & \text{wfCK}(F(\text{pl}), A(\text{pl}), K(\text{pl})) \\
& \{-3\} & s = ((\Diamond)(F(\text{pl}), \text{getExp}(\text{item1})) \cap (\Diamond)(F(\text{pl}), \text{getExp}(\text{item2}))) \\
& \{-4\} & \text{syntaxChangeCKLine}(K(\text{pl}), \text{ck2}, \text{item1}, \text{item2}, \text{its}) \\
& \{-5\} & \text{wt}(F(\text{pl}), \text{getExp}(\text{item2})) \\
\hline
& \{1\} & s(c) \Rightarrow (\text{prod}(\text{pl}, c) \text{ —- } \text{prod}((\#F := F(\text{pl}), A := A(\text{pl}), K := \text{ck2}\#), c))
\end{array}
$$

Applying disjunctive simplification to flatten sequent,

`changeCKLineStrongPartialRef.1:`

{-1}	$s(c) \Rightarrow$ ((FORALL (item: Item [Configuration, FeatureExpression, sat, FMi, Feature, [——], wf, wt, genFeatureExpression, getFeatures, addMandatory, addOptional]): union((items(K(pl)) \ items(ck2)), (items(ck2) \ items(K(pl)))) (item) $\Rightarrow \neg$ sat(getExp(item), c)) $\Rightarrow$ (semantics(K(pl))(A(pl))(c) = semantics(ck2)(A(pl))(c)))
{-2}	wfCK(F(pl), A(pl), K(pl))
{-3}	s = (($\diamond$)(F(pl), getExp(item1)) $\cap$ ($\diamond$)(F(pl), getExp(item2)))
{-4}	syntaxChangeCKLine(K(pl), ck2, item1, item2, its)
{-5}	wt(F(pl), getExp(item2))
{-6}	$s(c)$
{1}	(prod(pl, c) —- prod((#F := F(pl), A := A(pl), K := ck2#), c))

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

{-1}	$s(c)$
{-2}	(semantics(K(pl))(A(pl))(c) = semantics(ck2)(A(pl))(c))
{-3}	wfCK(F(pl), A(pl), K(pl))
{-4}	s = (($\diamond$)(F(pl), getExp(item1)) $\cap$ ($\diamond$)(F(pl), getExp(item2)))
{-5}	syntaxChangeCKLine(K(pl), ck2, item1, item2, its)
{-6}	wt(F(pl), getExp(item2))
{1}	(prod(pl, c) —- prod((#F := F(pl), A := A(pl), K := ck2#), c))

Expanding the definition of prod,
`changeCKLineStrongPartialRef.1.1:`

{-1}	$s(c)$
{-2}	(semantics(K(pl))(A(pl))(c) = semantics(ck2)(A(pl))(c))
{-3}	wfCK(F(pl), A(pl), K(pl))
{-4}	s = (($\diamond$)(F(pl), getExp(item1)) $\cap$ ($\diamond$)(F(pl), getExp(item2)))
{-5}	syntaxChangeCKLine(K(pl), ck2, item1, item2, its)
{-6}	wt(F(pl), getExp(item2))
{1}	((semantics(K(pl))(A(pl))(c)) —- (semantics(ck2)(A(pl))(c)))

Replacing using formula -2,

`changeCKLineStrongPartialRef.1.1:`

{-1} $s(c)$
{-2} (semantics(K(pl))(A(pl))(c) = semantics(ck2)(A(pl))(c))
{-3} wfCK(F(pl), A(pl), K(pl))
{-4} s = (($\diamond$)(F(pl), getExp(item1)) $\cap$ ($\diamond$)(F(pl), getExp(item2)))
{-5} syntaxChangeCKLine(K(pl), ck2, item1, item2, its)
{-6} wt(F(pl), getExp(item2))

{1} (semantics(ck2)(A(pl))(c) —- (semantics(ck2)(A(pl))(c)))

Using lemma SPLrefinement.assetRefinement,
`changeCKLineStrongPartialRef.1.1:`

{-1} orders[set[Asset]].preorder?(—-)
{-2} $s(c)$
{-3} (semantics(K(pl))(A(pl))(c) = semantics(ck2)(A(pl))(c))
{-4} wfCK(F(pl), A(pl), K(pl))
{-5} s = (($\diamond$)(F(pl), getExp(item1)) $\cap$ ($\diamond$)(F(pl), getExp(item2)))
{-6} syntaxChangeCKLine(K(pl), ck2, item1, item2, its)
{-7} wt(F(pl), getExp(item2))

{1} (semantics(ck2)(A(pl))(c) —- (semantics(ck2)(A(pl))(c)))

Expanding the definition of preorder?,
`changeCKLineStrongPartialRef.1.1:`

{-1} reflexive?(—-) & transitive?(—-)
{-2} $s(c)$
{-3} (semantics(K(pl))(A(pl))(c) = semantics(ck2)(A(pl))(c))
{-4} wfCK(F(pl), A(pl), K(pl))
{-5} s = (($\diamond$)(F(pl), getExp(item1)) $\cap$ ($\diamond$)(F(pl), getExp(item2)))
{-6} syntaxChangeCKLine(K(pl), ck2, item1, item2, its)
{-7} wt(F(pl), getExp(item2))

{1} (semantics(ck2)(A(pl))(c) —- (semantics(ck2)(A(pl))(c)))

Applying disjunctive simplification to flatten sequent,
`changeCKLineStrongPartialRef.1.1:`

{-1} reflexive?(—-)
{-2} transitive?(—-)
{-3} $s(c)$
{-4} (semantics(K(pl))(A(pl))(c) = semantics(ck2)(A(pl))(c))
{-5} wfCK(F(pl), A(pl), K(pl))
{-6} s = (($\diamond$)(F(pl), getExp(item1)) $\cap$ ($\diamond$)(F(pl), getExp(item2)))
{-7} syntaxChangeCKLine(K(pl), ck2, item1, item2, its)
{-8} wt(F(pl), getExp(item2))

{1} (semantics(ck2)(A(pl))(c) —- (semantics(ck2)(A(pl))(c)))

Expanding the definition of reflexive?,
`changeCKLineStrongPartialRef.1.1`:

{-1} FORALL (x: set[Asset]): (x —- x)
{-2} transitive?(—-)
{-3} $s(c)$
{-4} (semantics(K(pl))(A(pl))(c) = semantics(ck2)(A(pl))(c))
{-5} wfCK(F(pl), A(pl), K(pl))
{-6} s = ((◇)(F(pl), getExp(item1)) ∩ (◇)(F(pl), getExp(item2)))
{-7} syntaxChangeCKLine(K(pl), ck2, item1, item2, its)
{-8} wt(F(pl), getExp(item2))

{1} (semantics(ck2)(A(pl))(c) —- (semantics(ck2)(A(pl))(c)))

Instantiating the top quantifier in -1 with the terms: semantics(ck2)(A(pl))(c),
This completes the proof of `changeCKLineStrongPartialRef.1.1`.
`changeCKLineStrongPartialRef.1.2`:

{-1} $s(c)$
{-2} wfCK(F(pl), A(pl), K(pl))
{-3} s = ((◇)(F(pl), getExp(item1)) ∩ (◇)(F(pl), getExp(item2)))
{-4} syntaxChangeCKLine(K(pl), ck2, item1, item2, its)
{-5} wt(F(pl), getExp(item2))

{1} FORALL (item:
Item
[Configuration, FeatureExpression, sat, FMi, Feature, [——], wf, wt,
genFeatureExpression, getFeatures, addMandatory, addOptional]):
union((items(K(pl)) \ items(ck2)), (items(ck2) \ items(K(pl))))(item)
⇒ ¬ sat(getExp(item), c)
{2} (prod(pl, c) —- prod((#F := F(pl), A := A(pl), K := ck2#), c))

For the top quantifier in 1, we introduce Skolem constants: i,
`changeCKLineStrongPartialRef.1.2`:

{-1} $s(c)$
{-2} wfCK(F(pl), A(pl), K(pl))
{-3} s = ((◇)(F(pl), getExp(item1)) ∩ (◇)(F(pl), getExp(item2)))
{-4} syntaxChangeCKLine(K(pl), ck2, item1, item2, its)
{-5} wt(F(pl), getExp(item2))

{1} union((items(K(pl)) \ items(ck2)), (items(ck2) \ items(K(pl))))(i) ⇒
¬ sat(getExp(i), c)
{2} (prod(pl, c) —- prod((#F := F(pl), A := A(pl), K := ck2#), c))

Expanding the definition of union,

`changeCKLineStrongPartialRef.1.2:`

$$
\begin{array}{|ll}
\{\text{-}1\} & s(c) \\
\{\text{-}2\} & \mathrm{wfCK}(F(\mathrm{pl}),\ A(\mathrm{pl}),\ K(\mathrm{pl})) \\
\{\text{-}3\} & s\ =\ ((\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item1})) \cap (\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2}))) \\
\{\text{-}4\} & \mathrm{syntaxChangeCKLine}(K(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{item1},\ \mathrm{item2},\ \mathrm{its}) \\
\{\text{-}5\} & \mathrm{wt}(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2})) \\
\hline
\{1\} & (i \in (\mathrm{items}(K(\mathrm{pl})) \setminus \mathrm{items}(\mathrm{ck2})))\ \vee\ (i \in (\mathrm{items}(\mathrm{ck2}) \setminus \mathrm{items}(K(\mathrm{pl})))) \\
 & \quad \Rightarrow\ \neg\ \mathrm{sat}(\mathrm{getExp}(i),\ c) \\
\{2\} & (\mathrm{prod}(\mathrm{pl},\ c)\ \text{—-}\ \mathrm{prod}((\#F\ :=\ F(\mathrm{pl}),\ A\ :=\ A(\mathrm{pl}),\ K\ :=\ \mathrm{ck2}\#),\ c))
\end{array}
$$

Expanding the definition of member,
`changeCKLineStrongPartialRef.1.2:`

$$
\begin{array}{|ll}
\{\text{-}1\} & s(c) \\
\{\text{-}2\} & \mathrm{wfCK}(F(\mathrm{pl}),\ A(\mathrm{pl}),\ K(\mathrm{pl})) \\
\{\text{-}3\} & s\ =\ ((\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item1})) \cap (\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2}))) \\
\{\text{-}4\} & \mathrm{syntaxChangeCKLine}(K(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{item1},\ \mathrm{item2},\ \mathrm{its}) \\
\{\text{-}5\} & \mathrm{wt}(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2})) \\
\hline
\{1\} & \mathrm{difference}(\mathrm{items}(K(\mathrm{pl})),\ \mathrm{items}(\mathrm{ck2}))(i)\ \vee \\
 & \quad \mathrm{difference}(\mathrm{items}(\mathrm{ck2}),\ \mathrm{items}(K(\mathrm{pl})))(i) \\
 & \quad \Rightarrow\ \neg\ \mathrm{sat}(\mathrm{getExp}(i),\ c) \\
\{2\} & (\mathrm{prod}(\mathrm{pl},\ c)\ \text{—-}\ \mathrm{prod}((\#F\ :=\ F(\mathrm{pl}),\ A\ :=\ A(\mathrm{pl}),\ K\ :=\ \mathrm{ck2}\#),\ c))
\end{array}
$$

Expanding the definition of intersection,
`changeCKLineStrongPartialRef.1.2:`

$$
\begin{array}{|ll}
\{\text{-}1\} & s(c) \\
\{\text{-}2\} & \mathrm{wfCK}(F(\mathrm{pl}),\ A(\mathrm{pl}),\ K(\mathrm{pl})) \\
\{\text{-}3\} & s\ = \\
 & \quad (\{x\ \mid \\
 & \qquad\qquad (x \in (\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item1})))\ \wedge \\
 & \qquad\qquad\ (x \in (\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2})))\}) \\
\{\text{-}4\} & \mathrm{syntaxChangeCKLine}(K(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{item1},\ \mathrm{item2},\ \mathrm{its}) \\
\{\text{-}5\} & \mathrm{wt}(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2})) \\
\hline
\{1\} & \mathrm{difference}(\mathrm{items}(K(\mathrm{pl})),\ \mathrm{items}(\mathrm{ck2}))(i)\ \vee \\
 & \quad \mathrm{difference}(\mathrm{items}(\mathrm{ck2}),\ \mathrm{items}(K(\mathrm{pl})))(i) \\
 & \quad \Rightarrow\ \neg\ \mathrm{sat}(\mathrm{getExp}(i),\ c) \\
\{2\} & (\mathrm{prod}(\mathrm{pl},\ c)\ \text{—-}\ \mathrm{prod}((\#F\ :=\ F(\mathrm{pl}),\ A\ :=\ A(\mathrm{pl}),\ K\ :=\ \mathrm{ck2}\#),\ c))
\end{array}
$$

Expanding the definition of syntaxChangeCKLine,

`changeCKLineStrongPartialRef.1.2:`

$$
\begin{array}{ll}
\{\text{-}1\} & s(c) \\
\{\text{-}2\} & \mathrm{wfCK}(F(\mathrm{pl}),\ A(\mathrm{pl}),\ K(\mathrm{pl})) \\
\{\text{-}3\} & s\ = \\
 & \quad (\{x\ \mid \\
 & \qquad\qquad (x \in (\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item1})))\ \wedge \\
 & \qquad\qquad\ (x \in (\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2})))\}) \\
\{\text{-}4\} & \mathrm{items}(K(\mathrm{pl}))\ =\ (\mathrm{its} \cup \{\mathrm{item1}\})\ \wedge\ \mathrm{items}(\mathrm{ck2})\ =\ (\mathrm{its} \cup \{\mathrm{item2}\}) \\
\{\text{-}5\} & \mathrm{wt}(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2})) \\
\hline
\{1\} & \mathrm{difference}(\mathrm{items}(K(\mathrm{pl})),\ \mathrm{items}(\mathrm{ck2}))(i)\ \vee \\
 & \quad \mathrm{difference}(\mathrm{items}(\mathrm{ck2}),\ \mathrm{items}(K(\mathrm{pl})))(i) \\
 & \quad \Rightarrow\ \neg\ \mathrm{sat}(\mathrm{getExp}(i),\ c) \\
\{2\} & (\mathrm{prod}(\mathrm{pl},\ c)\ \text{—-}\ \mathrm{prod}((\#F\ :=\ F(\mathrm{pl}),\ A\ :=\ A(\mathrm{pl}),\ K\ :=\ \mathrm{ck2}\#),\ c))
\end{array}
$$

Expanding the definition of difference,

`changeCKLineStrongPartialRef.1.2:`

$$
\begin{array}{ll}
\{\text{-}1\} & s(c) \\
\{\text{-}2\} & \mathrm{wfCK}(F(\mathrm{pl}),\ A(\mathrm{pl}),\ K(\mathrm{pl})) \\
\{\text{-}3\} & s\ = \\
 & \quad (\{x\ \mid \\
 & \qquad\qquad (x \in (\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item1})))\ \wedge \\
 & \qquad\qquad\ (x \in (\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2})))\}) \\
\{\text{-}4\} & \mathrm{items}(K(\mathrm{pl}))\ =\ (\mathrm{its} \cup \{\mathrm{item1}\})\ \wedge\ \mathrm{items}(\mathrm{ck2})\ =\ (\mathrm{its} \cup \{\mathrm{item2}\}) \\
\{\text{-}5\} & \mathrm{wt}(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2})) \\
\hline
\{1\} & (i \in \mathrm{items}(K(\mathrm{pl})))\ \wedge\ \neg\ (i \in \mathrm{items}(\mathrm{ck2}))\ \vee \\
 & \quad (i \in \mathrm{items}(\mathrm{ck2}))\ \wedge\ \neg\ (i \in \mathrm{items}(K(\mathrm{pl}))) \\
 & \quad \Rightarrow\ \neg\ \mathrm{sat}(\mathrm{getExp}(i),\ c) \\
\{2\} & (\mathrm{prod}(\mathrm{pl},\ c)\ \text{—-}\ \mathrm{prod}((\#F\ :=\ F(\mathrm{pl}),\ A\ :=\ A(\mathrm{pl}),\ K\ :=\ \mathrm{ck2}\#),\ c))
\end{array}
$$

Expanding the definition of add,

`changeCKLineStrongPartialRef.1.2:`

$$
\begin{array}{l|ll}
 & \{-1\} & s(c) \\
 & \{-2\} & \mathrm{wfCK}(F(\mathrm{pl}),\ A(\mathrm{pl}),\ K(\mathrm{pl})) \\
 & \{-3\} & s\ = \\
 & & \quad (\{x\ \mid \\
 & & \qquad\qquad (x \in (\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item1}))) \ \wedge \\
 & & \qquad\qquad\ (x \in (\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2})))\}) \\
 & \{-4\} & (\mathrm{items}(K(\mathrm{pl}))\ =\ (\{y\ \mid\ \mathrm{item1}\ =\ y\ \vee\ (y \in \mathrm{its})\}))\ \wedge \\
 & & \quad \mathrm{items}(\mathrm{ck2})\ =\ (\{y\ \mid\ \mathrm{item2}\ =\ y\ \vee\ (y \in \mathrm{its})\}) \\
 & \{-5\} & \mathrm{wt}(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2})) \\
\hline
 & \{1\} & (i \in \mathrm{items}(K(\mathrm{pl})))\ \wedge\ \neg\ (i \in \mathrm{items}(\mathrm{ck2}))\ \vee \\
 & & \quad (i \in \mathrm{items}(\mathrm{ck2}))\ \wedge\ \neg\ (i \in \mathrm{items}(K(\mathrm{pl}))) \\
 & & \quad \Rightarrow\ \neg\ \mathrm{sat}(\mathrm{getExp}(i),\ c) \\
 & \{2\} & (\mathrm{prod}(\mathrm{pl},\ c)\ \text{—-}\ \mathrm{prod}((\#F\ :=\ F(\mathrm{pl}),\ A\ :=\ A(\mathrm{pl}),\ K\ :=\ \mathrm{ck2}\#),\ c))
\end{array}
$$

Expanding the definition of member,
`changeCKLineStrongPartialRef.1.2:`

$$
\begin{array}{l|ll}
 & \{-1\} & s(c) \\
 & \{-2\} & \mathrm{wfCK}(F(\mathrm{pl}),\ A(\mathrm{pl}),\ K(\mathrm{pl})) \\
 & \{-3\} & s\ = \\
 & & \quad (\{x\ \mid\ (\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item1}))(x)\ \wedge\ (\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2}))(x)\}) \\
 & \{-4\} & (\mathrm{items}(K(\mathrm{pl}))\ =\ (\{y\ \mid\ \mathrm{item1}\ =\ y\ \vee\ \mathrm{its}(y)\}))\ \wedge \\
 & & \quad \mathrm{items}(\mathrm{ck2})\ =\ (\{y\ \mid\ \mathrm{item2}\ =\ y\ \vee\ \mathrm{its}(y)\}) \\
 & \{-5\} & \mathrm{wt}(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2})) \\
\hline
 & \{1\} & \mathrm{items}(K(\mathrm{pl}))(i)\ \wedge\ \neg\ \mathrm{items}(\mathrm{ck2})(i)\ \vee \\
 & & \quad \mathrm{items}(\mathrm{ck2})(i)\ \wedge\ \neg\ \mathrm{items}(K(\mathrm{pl}))(i) \\
 & & \quad \Rightarrow\ \neg\ \mathrm{sat}(\mathrm{getExp}(i),\ c) \\
 & \{2\} & (\mathrm{prod}(\mathrm{pl},\ c)\ \text{—-}\ \mathrm{prod}((\#F\ :=\ F(\mathrm{pl}),\ A\ :=\ A(\mathrm{pl}),\ K\ :=\ \mathrm{ck2}\#),\ c))
\end{array}
$$

Applying disjunctive simplification to flatten sequent,
`changeCKLineStrongPartialRef.1.2:`

$$
\begin{array}{l|ll}
 & \{-1\} & s(c) \\
 & \{-2\} & \mathrm{items}(K(\mathrm{pl}))(i)\ \wedge\ \neg\ \mathrm{items}(\mathrm{ck2})(i)\ \vee\ \mathrm{items}(\mathrm{ck2})(i)\ \wedge\ \neg\ \mathrm{items}(K(\mathrm{pl}))(i) \\
 & \{-3\} & \mathrm{sat}(\mathrm{getExp}(i),\ c) \\
 & \{-4\} & \mathrm{wfCK}(F(\mathrm{pl}),\ A(\mathrm{pl}),\ K(\mathrm{pl})) \\
 & \{-5\} & s\ = \\
 & & \quad (\{x\ \mid\ (\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item1}))(x)\ \wedge\ (\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2}))(x)\}) \\
 & \{-6\} & \mathrm{items}(K(\mathrm{pl}))\ =\ (\{y\ \mid\ \mathrm{item1}\ =\ y\ \vee\ \mathrm{its}(y)\}) \\
 & \{-7\} & \mathrm{items}(\mathrm{ck2})\ =\ (\{y\ \mid\ \mathrm{item2}\ =\ y\ \vee\ \mathrm{its}(y)\}) \\
 & \{-8\} & \mathrm{wt}(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2})) \\
\hline
 & \{1\} & (\mathrm{prod}(\mathrm{pl},\ c)\ \text{—-}\ \mathrm{prod}((\#F\ :=\ F(\mathrm{pl}),\ A\ :=\ A(\mathrm{pl}),\ K\ :=\ \mathrm{ck2}\#),\ c))
\end{array}
$$

Applying decompose-equality,

`changeCKLineStrongPartialRef.1.2:`

{-1} ∀ (x:
Item
[Configuration, FeatureExpression, sat, FMi, Feature, [——], wf, wt,
genFeatureExpression, getFeatures, addMandatory, addOptional]):
items(K(pl))(x) = (item1 = x ∨ its(x))
{-2} $s(c)$
{-3} items(K(pl))(i) ∧ ¬ items(ck2)(i) ∨ items(ck2)(i) ∧ ¬ items(K(pl))(i)
{-4} sat(getExp(i), c)
{-5} wfCK(F(pl), A(pl), K(pl))
{-6} s =
({x | (◇)(F(pl), getExp(item1))(x) ∧ (◇)(F(pl), getExp(item2))(x)})
{-7} items(ck2) = ({y | item2 = y ∨ its(y)})
{-8} wt(F(pl), getExp(item2))

{1} (prod(pl, c) —- prod((#F := F(pl), A := A(pl), K := ck2#), c))

Applying decompose-equality,
`changeCKLineStrongPartialRef.1.2:`

{-1} ∀ (x:
Item
[Configuration, FeatureExpression, sat, FMi, Feature, [——], wf, wt,
genFeatureExpression, getFeatures, addMandatory, addOptional]):
items(ck2)(x) = (item2 = x ∨ its(x))
{-2} ∀ (x:
Item
[Configuration, FeatureExpression, sat, FMi, Feature, [——], wf, wt,
genFeatureExpression, getFeatures, addMandatory, addOptional]):
items(K(pl))(x) = (item1 = x ∨ its(x))
{-3} $s(c)$
{-4} items(K(pl))(i) ∧ ¬ items(ck2)(i) ∨ items(ck2)(i) ∧ ¬ items(K(pl))(i)
{-5} sat(getExp(i), c)
{-6} wfCK(F(pl), A(pl), K(pl))
{-7} s =
({x | (◇)(F(pl), getExp(item1))(x) ∧ (◇)(F(pl), getExp(item2))(x)})
{-8} wt(F(pl), getExp(item2))

{1} (prod(pl, c) —- prod((#F := F(pl), A := A(pl), K := ck2#), c))

Instantiating the top quantifier in -1 with the terms: i,
`changeCKLineStrongPartialRef.1.2`:

$$
\begin{array}{|ll}
\{-1\} & \text{items}(\text{ck2})(i) = (\text{item2} = i \vee \text{its}(i)) \\
\{-2\} & \forall\ (x: \\
& \qquad \text{Item} \\
& \qquad\quad [\text{Configuration},\ \text{FeatureExpression},\ \text{sat},\ \text{FMi},\ \text{Fea-} \\
& \text{ture},\ [\text{——}],\ \text{wf},\ \text{wt}, \\
& \qquad\quad \text{genFeatureExpression},\ \text{getFeatures},\ \text{addMandatory},\ \text{ad-} \\
& \text{dOptional}]): \\
& \quad \text{items}(K(\text{pl}))(x) = (\text{item1} = x \vee \text{its}(x)) \\
\{-3\} & s(c) \\
\{-4\} & \text{items}(K(\text{pl}))(i) \wedge \neg\ \text{items}(\text{ck2})(i) \vee \text{items}(\text{ck2})(i) \wedge \neg\ \text{items}(K(\text{pl}))(i) \\
\{-5\} & \text{sat}(\text{getExp}(i),\ c) \\
\{-6\} & \text{wfCK}(F(\text{pl}),\ A(\text{pl}),\ K(\text{pl})) \\
\{-7\} & s = \\
& \quad (\{x \mid (\diamond)(F(\text{pl}),\ \text{getExp}(\text{item1}))(x) \wedge (\diamond)(F(\text{pl}),\ \text{getExp}(\text{item2}))(x)\}) \\
\{-8\} & \text{wt}(F(\text{pl}),\ \text{getExp}(\text{item2})) \\
\hline
\{1\} & (\text{prod}(\text{pl},\ c) \text{ —- } \text{prod}((\#F := F(\text{pl}),\ A := A(\text{pl}),\ K := \text{ck2}\#),\ c))
\end{array}
$$

Instantiating the top quantifier in -2 with the terms: i,
`changeCKLineStrongPartialRef.1.2`:

$$
\begin{array}{|ll}
\{-1\} & \text{items}(\text{ck2})(i) = (\text{item2} = i \vee \text{its}(i)) \\
\{-2\} & \text{items}(K(\text{pl}))(i) = (\text{item1} = i \vee \text{its}(i)) \\
\{-3\} & s(c) \\
\{-4\} & \text{items}(K(\text{pl}))(i) \wedge \neg\ \text{items}(\text{ck2})(i) \vee \text{items}(\text{ck2})(i) \wedge \neg\ \text{items}(K(\text{pl}))(i) \\
\{-5\} & \text{sat}(\text{getExp}(i),\ c) \\
\{-6\} & \text{wfCK}(F(\text{pl}),\ A(\text{pl}),\ K(\text{pl})) \\
\{-7\} & s = \\
& \quad (\{x \mid (\diamond)(F(\text{pl}),\ \text{getExp}(\text{item1}))(x) \wedge (\diamond)(F(\text{pl}),\ \text{getExp}(\text{item2}))(x)\}) \\
\{-8\} & \text{wt}(F(\text{pl}),\ \text{getExp}(\text{item2})) \\
\hline
\{1\} & (\text{prod}(\text{pl},\ c) \text{ —- } \text{prod}((\#F := F(\text{pl}),\ A := A(\text{pl}),\ K := \text{ck2}\#),\ c))
\end{array}
$$

Applying bddsimp,
we get 2 subgoals:

`changeCKLineStrongPartialRef.1.2.1:`

{-1}	items(ck2)(i)
{-2}	item2 $=$ i
{-3}	$s(c)$
{-4}	sat(getExp(i), c)
{-5}	wfCK(F(pl), A(pl), K(pl))
{-6}	s $=$ ({x | ($\Diamond$)(F(pl), getExp(item1))(x) $\wedge$ ($\Diamond$)(F(pl), getExp(item2))(x)})
{-7}	wt(F(pl), getExp(item2))
{1}	its(i)
{2}	items(K(pl))(i)
{3}	item1 $=$ i
{4}	(prod(pl, c) —- prod((#F := F(pl), A := A(pl), K := ck2#), c))

Applying decompose-equality,
`changeCKLineStrongPartialRef.1.2.1:`

{-1}	$\forall$ (x_1: Configuration): $s(x_1)$ $=$ (($\Diamond$)(F(pl), getExp(item1))(x_1) $\wedge$ ($\Diamond$)(F(pl), getExp(item2))(x_1))
{-2}	items(ck2)(i)
{-3}	item2 $=$ i
{-4}	$s(c)$
{-5}	sat(getExp(i), c)
{-6}	wfCK(F(pl), A(pl), K(pl))
{-7}	wt(F(pl), getExp(item2))
{1}	its(i)
{2}	items(K(pl))(i)
{3}	item1 $=$ i
{4}	(prod(pl, c) —- prod((#F := F(pl), A := A(pl), K := ck2#), c))

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

`changeCKLineStrongPartialRef.1.2.1:`

{-1}	$s(c) =$ $((\Diamond)(F(\text{pl}),\ \text{getExp}(\text{item1}))(c)\ \wedge\ (\Diamond)(F(\text{pl}),\ \text{getExp}(\text{item2}))(c))$
{-2}	$\text{items}(\text{ck2})(i)$
{-3}	$\text{item2}\ =\ i$
{-4}	$s(c)$
{-5}	$\text{sat}(\text{getExp}(i),\ c)$
{-6}	$\text{wfCK}(F(\text{pl}),\ A(\text{pl}),\ K(\text{pl}))$
{-7}	$\text{wt}(F(\text{pl}),\ \text{getExp}(\text{item2}))$
{1}	$\text{its}(i)$
{2}	$\text{items}(K(\text{pl}))(i)$
{3}	$\text{item1}\ =\ i$
{4}	$(\text{prod}(\text{pl},\ c)$ —- $\text{prod}((\#F\ :=\ F(\text{pl}),\ A\ :=\ A(\text{pl}),\ K\ :=\ \text{ck2}\#),\ c))$

Expanding the definition of ¡¿,
`changeCKLineStrongPartialRef.1.2.1:`

{-1}	$s(c) =$ $(([\text{——}](F(\text{pl}))(c)\ \wedge\ \neg\ \text{sat}(\text{getExp}(\text{item1}),\ c))\ \wedge$ $[\text{——}](F(\text{pl}))(c)\ \wedge\ \neg\ \text{sat}(\text{getExp}(\text{item2}),\ c))$
{-2}	$\text{items}(\text{ck2})(i)$
{-3}	$\text{item2}\ =\ i$
{-4}	$s(c)$
{-5}	$\text{sat}(\text{getExp}(i),\ c)$
{-6}	$\text{wfCK}(F(\text{pl}),\ A(\text{pl}),\ K(\text{pl}))$
{-7}	$\text{wt}(F(\text{pl}),\ \text{getExp}(\text{item2}))$
{1}	$\text{its}(i)$
{2}	$\text{items}(K(\text{pl}))(i)$
{3}	$\text{item1}\ =\ i$
{4}	$(\text{prod}(\text{pl},\ c)$ —- $\text{prod}((\#F\ :=\ F(\text{pl}),\ A\ :=\ A(\text{pl}),\ K\ :=\ \text{ck2}\#),\ c))$

Applying bddsimp,

`changeCKLineStrongPartialRef.1.2.1`:

{-1}	$s(c)$
{-2}	[——](F(pl))(c)
{-3}	items(ck2)(i)
{-4}	item2 $=$ i
{-5}	sat(getExp(i), c)
{-6}	wfCK(F(pl), A(pl), K(pl))
{-7}	wt(F(pl), getExp(item2))
{1}	sat(getExp(item1), c)
{2}	sat(getExp(item2), c)
{3}	its(i)
{4}	items(K(pl))(i)
{5}	item1 $=$ i
{6}	(prod(pl, c) —- prod((#F := F(pl), A := A(pl), K := ck2#), c))

Replacing using formula -4,
`changeCKLineStrongPartialRef.1.2.1`:

{-1}	$s(c)$
{-2}	[——](F(pl))(c)
{-3}	items(ck2)(i)
{-4}	item2 $=$ i
{-5}	sat(getExp(i), c)
{-6}	wfCK(F(pl), A(pl), K(pl))
{-7}	wt(F(pl), getExp(item2))
{1}	sat(getExp(item1), c)
{2}	sat(getExp(i), c)
{3}	its(i)
{4}	items(K(pl))(i)
{5}	item1 $=$ i
{6}	(prod(pl, c) —- prod((#F := F(pl), A := A(pl), K := ck2#), c))

which is trivially true.
This completes the proof of `changeCKLineStrongPartialRef.1.2.1`.

`changeCKLineStrongPartialRef.1.2.2:`

{-1}	items(K(pl))(i)
{-2}	item1 = i
{-3}	s(c)
{-4}	sat(getExp(i), c)
{-5}	wfCK(F(pl), A(pl), K(pl))
{-6}	s = ({x | (◇)(F(pl), getExp(item1))(x) ∧ (◇)(F(pl), getExp(item2))(x)})
{-7}	wt(F(pl), getExp(item2))
{1}	items(ck2)(i)
{2}	item2 = i
{3}	its(i)
{4}	(prod(pl, c) —- prod((#F := F(pl), A := A(pl), K := ck2#), c))

Applying decompose-equality,

`changeCKLineStrongPartialRef.1.2.2:`

{-1}	∀ (x_1: Configuration): $s(x_1)$ = ((◇)(F(pl), getExp(item1))(x_1) ∧ (◇)(F(pl), getExp(item2))(x_1))
{-2}	items(K(pl))(i)
{-3}	item1 = i
{-4}	s(c)
{-5}	sat(getExp(i), c)
{-6}	wfCK(F(pl), A(pl), K(pl))
{-7}	wt(F(pl), getExp(item2))
{1}	items(ck2)(i)
{2}	item2 = i
{3}	its(i)
{4}	(prod(pl, c) —- prod((#F := F(pl), A := A(pl), K := ck2#), c))

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

`changeCKLineStrongPartialRef.1.2.2:`

{-1}	$s(c) =$ $((\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item1}))(c) \wedge (\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2}))(c))$
{-2}	$\mathrm{items}(K(\mathrm{pl}))(i)$
{-3}	$\mathrm{item1} = i$
{-4}	$s(c)$
{-5}	$\mathrm{sat}(\mathrm{getExp}(i),\ c)$
{-6}	$\mathrm{wfCK}(F(\mathrm{pl}),\ A(\mathrm{pl}),\ K(\mathrm{pl}))$
{-7}	$\mathrm{wt}(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2}))$
{1}	$\mathrm{items}(\mathrm{ck2})(i)$
{2}	$\mathrm{item2} = i$
{3}	$\mathrm{its}(i)$
{4}	$(\mathrm{prod}(\mathrm{pl},\ c)$ —- $\mathrm{prod}((\#F := F(\mathrm{pl}),\ A := A(\mathrm{pl}),\ K := \mathrm{ck2}\#),\ c))$

Expanding the definition of ¡¿,
`changeCKLineStrongPartialRef.1.2.2:`

{-1}	$s(c) =$ $(([\text{——}](F(\mathrm{pl}))(c) \wedge \neg\ \mathrm{sat}(\mathrm{getExp}(\mathrm{item1}),\ c)) \wedge$ $[\text{——}](F(\mathrm{pl}))(c) \wedge \neg\ \mathrm{sat}(\mathrm{getExp}(\mathrm{item2}),\ c))$
{-2}	$\mathrm{items}(K(\mathrm{pl}))(i)$
{-3}	$\mathrm{item1} = i$
{-4}	$s(c)$
{-5}	$\mathrm{sat}(\mathrm{getExp}(i),\ c)$
{-6}	$\mathrm{wfCK}(F(\mathrm{pl}),\ A(\mathrm{pl}),\ K(\mathrm{pl}))$
{-7}	$\mathrm{wt}(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2}))$
{1}	$\mathrm{items}(\mathrm{ck2})(i)$
{2}	$\mathrm{item2} = i$
{3}	$\mathrm{its}(i)$
{4}	$(\mathrm{prod}(\mathrm{pl},\ c)$ —- $\mathrm{prod}((\#F := F(\mathrm{pl}),\ A := A(\mathrm{pl}),\ K := \mathrm{ck2}\#),\ c))$

Applying bddsimp,

`changeCKLineStrongPartialRef.1.2.2`:

{-1}	$s(c)$
{-2}	$[\text{——}](F(\text{pl}))(c)$
{-3}	$\text{items}(K(\text{pl}))(i)$
{-4}	$\text{item1} = i$
{-5}	$\text{sat}(\text{getExp}(i),\ c)$
{-6}	$\text{wfCK}(F(\text{pl}),\ A(\text{pl}),\ K(\text{pl}))$
{-7}	$\text{wt}(F(\text{pl}),\ \text{getExp}(\text{item2}))$
{1}	$\text{sat}(\text{getExp}(\text{item1}),\ c)$
{2}	$\text{sat}(\text{getExp}(\text{item2}),\ c)$
{3}	$\text{items}(\text{ck2})(i)$
{4}	$\text{item2} = i$
{5}	$\text{its}(i)$
{6}	$(\text{prod}(\text{pl},\ c)\ \text{—-}\ \text{prod}((\#F := F(\text{pl}),\ A := A(\text{pl}),\ K := \text{ck2}\#),\ c))$

Replacing using formula -4,
`changeCKLineStrongPartialRef.1.2.2`:

{-1}	$s(c)$
{-2}	$[\text{——}](F(\text{pl}))(c)$
{-3}	$\text{items}(K(\text{pl}))(i)$
{-4}	$\text{item1} = i$
{-5}	$\text{sat}(\text{getExp}(i),\ c)$
{-6}	$\text{wfCK}(F(\text{pl}),\ A(\text{pl}),\ K(\text{pl}))$
{-7}	$\text{wt}(F(\text{pl}),\ \text{getExp}(\text{item2}))$
{1}	$\text{sat}(\text{getExp}(i),\ c)$
{2}	$\text{sat}(\text{getExp}(\text{item2}),\ c)$
{3}	$\text{items}(\text{ck2})(i)$
{4}	$\text{item2} = i$
{5}	$\text{its}(i)$
{6}	$(\text{prod}(\text{pl},\ c)\ \text{—-}\ \text{prod}((\#F := F(\text{pl}),\ A := A(\text{pl}),\ K := \text{ck2}\#),\ c))$

which is trivially true.
This completes the proof of `changeCKLineStrongPartialRef.1.2.2`.
`changeCKLineStrongPartialRef.2`:

{-1}	$\text{wfCK}(F(\text{pl}),\ A(\text{pl}),\ K(\text{pl}))$
{-2}	$s = ((\Diamond)(F(\text{pl}),\ \text{getExp}(\text{item1})) \cap (\Diamond)(F(\text{pl}),\ \text{getExp}(\text{item2})))$
{-3}	$\text{syntaxChangeCKLine}(K(\text{pl}),\ \text{ck2},\ \text{item1},\ \text{item2},\ \text{its})$
{-4}	$\text{wt}(F(\text{pl}),\ \text{getExp}(\text{item2}))$
{1}	$(s \subseteq [\text{——}](F(\text{pl})))$

Expanding the definition of subset?,

`changeCKLineStrongPartialRef.2:`

$$
\begin{array}{l|ll}
& \{\text{-}1\} & \mathrm{wfCK}(F(\mathrm{pl}),\ A(\mathrm{pl}),\ K(\mathrm{pl})) \\
& \{\text{-}2\} & s\ =\ ((\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item1})) \cap (\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2}))) \\
& \{\text{-}3\} & \mathrm{syntaxChangeCKLine}(K(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{item1},\ \mathrm{item2},\ \mathrm{its}) \\
& \{\text{-}4\} & \mathrm{wt}(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2})) \\
\hline
& \{1\} & \textsc{forall}\ (x\colon \mathrm{Configuration})\colon\ (x \in s)\ \Rightarrow\ (x \in [\text{——}](F(\mathrm{pl})))
\end{array}
$$

Expanding the definition of intersection,
`changeCKLineStrongPartialRef.2:`

$$
\begin{array}{l|ll}
& \{\text{-}1\} & \mathrm{wfCK}(F(\mathrm{pl}),\ A(\mathrm{pl}),\ K(\mathrm{pl})) \\
& \{\text{-}2\} & s\ = \\
& & \quad (\{x\ \mid \\
& & \qquad\qquad (x \in (\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item1})))\ \wedge \\
& & \qquad\qquad\ (x \in (\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2})))\}) \\
& \{\text{-}3\} & \mathrm{syntaxChangeCKLine}(K(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{item1},\ \mathrm{item2},\ \mathrm{its}) \\
& \{\text{-}4\} & \mathrm{wt}(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2})) \\
\hline
& \{1\} & \textsc{forall}\ (x\colon \mathrm{Configuration})\colon\ (x \in s)\ \Rightarrow\ (x \in [\text{——}](F(\mathrm{pl})))
\end{array}
$$

Applying decompose-equality,
`changeCKLineStrongPartialRef.2:`

$$
\begin{array}{l|ll}
& \{\text{-}1\} & \forall\ (x_1\colon\ \mathrm{Configuration})\colon \\
& & \qquad s(x_1)\ = \\
& & \qquad\ ((x_1 \in (\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item1})))\ \wedge \\
& & \qquad\qquad (x_1 \in (\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2})))) \\
& \{\text{-}2\} & \mathrm{wfCK}(F(\mathrm{pl}),\ A(\mathrm{pl}),\ K(\mathrm{pl})) \\
& \{\text{-}3\} & \mathrm{syntaxChangeCKLine}(K(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{item1},\ \mathrm{item2},\ \mathrm{its}) \\
& \{\text{-}4\} & \mathrm{wt}(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2})) \\
\hline
& \{1\} & \textsc{forall}\ (x\colon \mathrm{Configuration})\colon\ (x \in s)\ \Rightarrow\ (x \in [\text{——}](F(\mathrm{pl})))
\end{array}
$$

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

$$
\begin{array}{l|ll}
& \{\text{-}1\} & \forall\ (x_1\colon\ \mathrm{Configuration})\colon \\
& & \qquad s(x_1)\ = \\
& & \qquad\ ((x_1 \in (\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item1})))\ \wedge \\
& & \qquad\qquad (x_1 \in (\Diamond)(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2})))) \\
& \{\text{-}2\} & \mathrm{wfCK}(F(\mathrm{pl}),\ A(\mathrm{pl}),\ K(\mathrm{pl})) \\
& \{\text{-}3\} & \mathrm{syntaxChangeCKLine}(K(\mathrm{pl}),\ \mathrm{ck2},\ \mathrm{item1},\ \mathrm{item2},\ \mathrm{its}) \\
& \{\text{-}4\} & \mathrm{wt}(F(\mathrm{pl}),\ \mathrm{getExp}(\mathrm{item2})) \\
\hline
& \{1\} & (c \in s)\ \Rightarrow\ (c \in [\text{——}](F(\mathrm{pl})))
\end{array}
$$

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

`changeCKLineStrongPartialRef.2:`

$$
\begin{array}{l|ll}
 & \{\text{-}1\} & s(c) \ = \\
 & & \quad ((c \in (\Diamond)(F(\text{pl}),\ \text{getExp}(\text{item1})))\ \wedge \\
 & & \qquad (c \in (\Diamond)(F(\text{pl}),\ \text{getExp}(\text{item2})))) \\
 & \{\text{-}2\} & \text{wfCK}(F(\text{pl}),\ A(\text{pl}),\ K(\text{pl})) \\
 & \{\text{-}3\} & \text{syntaxChangeCKLine}(K(\text{pl}),\ \text{ck2},\ \text{item1},\ \text{item2},\ \text{its}) \\
 & \{\text{-}4\} & \text{wt}(F(\text{pl}),\ \text{getExp}(\text{item2})) \\
\hline
 & \{1\} & (c \in s)\ \Rightarrow\ (c \in [\text{——}](F(\text{pl})))
\end{array}
$$

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

$$
\begin{array}{l|ll}
 & \{\text{-}1\} & s(c) \ = \\
 & & \quad ((\Diamond)(F(\text{pl}),\ \text{getExp}(\text{item1}))(c)\ \wedge\ (\Diamond)(F(\text{pl}),\ \text{getExp}(\text{item2}))(c)) \\
 & \{\text{-}2\} & \text{wfCK}(F(\text{pl}),\ A(\text{pl}),\ K(\text{pl})) \\
 & \{\text{-}3\} & \text{syntaxChangeCKLine}(K(\text{pl}),\ \text{ck2},\ \text{item1},\ \text{item2},\ \text{its}) \\
 & \{\text{-}4\} & \text{wt}(F(\text{pl}),\ \text{getExp}(\text{item2})) \\
\hline
 & \{1\} & s(c)\ \Rightarrow\ [\text{——}](F(\text{pl}))(c)
\end{array}
$$

Expanding the definition of ¡¿,
`changeCKLineStrongPartialRef.2:`

$$
\begin{array}{l|ll}
 & \{\text{-}1\} & s(c) \ = \\
 & & \quad (([\text{——}](F(\text{pl}))(c)\ \wedge\ \neg\ \text{sat}(\text{getExp}(\text{item1}),\ c))\ \wedge \\
 & & \qquad [\text{——}](F(\text{pl}))(c)\ \wedge\ \neg\ \text{sat}(\text{getExp}(\text{item2}),\ c)) \\
 & \{\text{-}2\} & \text{wfCK}(F(\text{pl}),\ A(\text{pl}),\ K(\text{pl})) \\
 & \{\text{-}3\} & \text{syntaxChangeCKLine}(K(\text{pl}),\ \text{ck2},\ \text{item1},\ \text{item2},\ \text{its}) \\
 & \{\text{-}4\} & \text{wt}(F(\text{pl}),\ \text{getExp}(\text{item2})) \\
\hline
 & \{1\} & s(c)\ \Rightarrow\ [\text{——}](F(\text{pl}))(c)
\end{array}
$$

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

Verbose proof for `removeFeaturePartRefStrong`.
`removeFeaturePartRefStrong`:

{1}	FORALL (pl, pl2, s, its, pairs, P, Q): (predRemoveFeature(pl, pl2, s, its, pairs, P, Q) $\Rightarrow$ strongPartialRefinement(pl, pl2, s))

`removeFeaturePartRefStrong`:

{1}	FORALL (pl, pl2, s, its, pairs, P, Q): (predRemoveFeature(pl, pl2, s, its, pairs, P, Q) $\Rightarrow$ strongPartialRefinement(pl, pl2, s))

For the top quantifier in 1, we introduce Skolem constants: (pl pl2 s its pairs P Q),
`removeFeaturePartRefStrong`:

{1}	(predRemoveFeature(pl, pl2, s, its, pairs, P, Q) $\Rightarrow$ strongPartialRefinement(pl, pl2, s))

Applying removeFeatureSameProducts
`removeFeaturePartRefStrong`:

{-1}	$\forall$ (pl, pl2, s, its, pairs, P, Q): (predRemoveFeature(pl, pl2, s, its, pairs, P, Q) $\Rightarrow$ (FORALL c: $s(c)$ $\Rightarrow$ prod(pl, c) = prod(pl2, c)))
{1}	(predRemoveFeature(pl, pl2, s, its, pairs, P, Q) $\Rightarrow$ strongPartialRefinement(pl, pl2, s))

Instantiating the top quantifier in -1 with the terms: pl, pl2, s, its, pairs, P, Q,
`removeFeaturePartRefStrong`:

{-1}	(predRemoveFeature(pl, pl2, s, its, pairs, P, Q) $\Rightarrow$ (FORALL c: $s(c)$ $\Rightarrow$ prod(pl, c) = prod(pl2, c)))
{1}	(predRemoveFeature(pl, pl2, s, its, pairs, P, Q) $\Rightarrow$ strongPartialRefinement(pl, pl2, s))

Applying bddsimp,
`removeFeaturePartRefStrong`:

{-1}	predRemoveFeature(pl, pl2, s, its, pairs, P, Q)
{-2}	FORALL c: $s(c)$ $\Rightarrow$ prod(pl, c) = prod(pl2, c)
{1}	strongPartialRefinement(pl, pl2, s)

Expanding the definition of strongPartialRefinement, `removeFeaturePartRefStrong`:

{-1} predRemoveFeature(pl, pl2, s, its, pairs, P, Q)
{-2} FORALL c: $s(c)$ ⇒ prod(pl, c) = prod(pl2, c)

{1} ($s \subseteq$ semantics(F(pl))) ∧
($s \subseteq$ semantics(F(pl2))) ∧
(FORALL (c: Configuration): $s(c)$ ⇒ (prod(pl, c) —- prod(pl2, c)))

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

{-1} predRemoveFeature(pl, pl2, s, its, pairs, P, Q)
{-2} FORALL c: $s(c)$ ⇒ prod(pl, c) = prod(pl2, c)

{1} FORALL (c: Configuration): $s(c)$ ⇒ (prod(pl, c) —- prod(pl2, c))

Using lemma SPLrefinement.assetRefinement,
`removeFeaturePartRefStrong.1`:

{-1} orders[set[Assets.Asset]].preorder?(—-)
{-2} predRemoveFeature(pl, pl2, s, its, pairs, P, Q)
{-3} FORALL c: $s(c)$ ⇒ prod(pl, c) = prod(pl2, c)

{1} FORALL (c: Configuration): $s(c)$ ⇒ (prod(pl, c) —- prod(pl2, c))

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

{-1} reflexive?(—-) & transitive?(—-)
{-2} predRemoveFeature(pl, pl2, s, its, pairs, P, Q)
{-3} FORALL c: $s(c)$ ⇒ prod(pl, c) = prod(pl2, c)

{1} FORALL (c: Configuration): $s(c)$ ⇒ (prod(pl, c) —- prod(pl2, c))

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

{-1} reflexive?(—-)
{-2} transitive?(—-)
{-3} predRemoveFeature(pl, pl2, s, its, pairs, P, Q)
{-4} FORALL c: $s(c)$ ⇒ prod(pl, c) = prod(pl2, c)

{1} FORALL (c: Configuration): $s(c)$ ⇒ (prod(pl, c) —- prod(pl2, c))

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

{-1} FORALL (x: set[Assets.Asset]): (x —- x)
{-2} transitive?(—-)
{-3} predRemoveFeature(pl, pl2, s, its, pairs, P, Q)
{-4} FORALL c: $s(c)$ ⇒ prod(pl, c) = prod(pl2, c)

{1} FORALL (c: Configuration): $s(c)$ ⇒ (prod(pl, c) —- prod(pl2, c))

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

$$
\begin{array}{ll}
\{\text{-}1\} & \textsc{forall}\ (x\colon \text{set}[\text{Assets.Asset}])\colon (x \text{ —- } x) \\
\{\text{-}2\} & \text{transitive?(—-)} \\
\{\text{-}3\} & \text{predRemoveFeature}(\text{pl},\ \text{pl2},\ s,\ \text{its},\ \text{pairs},\ P,\ Q) \\
\{\text{-}4\} & \textsc{forall}\ c\colon\ s(c) \Rightarrow \text{prod}(\text{pl},\ c) = \text{prod}(\text{pl2},\ c) \\
\hline
\{1\} & s(c) \Rightarrow (\text{prod}(\text{pl},\ c) \text{ —- } \text{prod}(\text{pl2},\ c))
\end{array}
$$

Instantiating the top quantifier in -4 with the terms: c,
`removeFeaturePartRefStrong.1:`

$$
\begin{array}{ll}
\{\text{-}1\} & \textsc{forall}\ (x\colon \text{set}[\text{Assets.Asset}])\colon (x \text{ —- } x) \\
\{\text{-}2\} & \text{transitive?(—-)} \\
\{\text{-}3\} & \text{predRemoveFeature}(\text{pl},\ \text{pl2},\ s,\ \text{its},\ \text{pairs},\ P,\ Q) \\
\{\text{-}4\} & s(c) \Rightarrow \text{prod}(\text{pl},\ c) = \text{prod}(\text{pl2},\ c) \\
\hline
\{1\} & s(c) \Rightarrow (\text{prod}(\text{pl},\ c) \text{ —- } \text{prod}(\text{pl2},\ c))
\end{array}
$$

Applying bddsimp,
`removeFeaturePartRefStrong.1:`

$$
\begin{array}{ll}
\{\text{-}1\} & \textsc{forall}\ (x\colon \text{set}[\text{Assets.Asset}])\colon (x \text{ —- } x) \\
\{\text{-}2\} & \text{transitive?(—-)} \\
\{\text{-}3\} & \text{predRemoveFeature}(\text{pl},\ \text{pl2},\ s,\ \text{its},\ \text{pairs},\ P,\ Q) \\
\{\text{-}4\} & s(c) \\
\{\text{-}5\} & \text{prod}(\text{pl},\ c) = \text{prod}(\text{pl2},\ c) \\
\hline
\{1\} & (\text{prod}(\text{pl},\ c) \text{ —- } \text{prod}(\text{pl2},\ c))
\end{array}
$$

Replacing using formula -5,
`removeFeaturePartRefStrong.1:`

$$
\begin{array}{ll}
\{\text{-}1\} & \textsc{forall}\ (x\colon \text{set}[\text{Assets.Asset}])\colon (x \text{ —- } x) \\
\{\text{-}2\} & \text{transitive?(—-)} \\
\{\text{-}3\} & \text{predRemoveFeature}(\text{pl},\ \text{pl2},\ s,\ \text{its},\ \text{pairs},\ P,\ Q) \\
\{\text{-}4\} & s(c) \\
\{\text{-}5\} & \text{prod}(\text{pl},\ c) = \text{prod}(\text{pl2},\ c) \\
\hline
\{1\} & (\text{prod}(\text{pl2},\ c) \text{ —- } \text{prod}(\text{pl2},\ c))
\end{array}
$$

Instantiating the top quantifier in -1 with the terms: prod(pl2, c),
This completes the proof of `removeFeaturePartRefStrong.1`.
`removeFeaturePartRefStrong.2:`

$$
\begin{array}{ll}
\{\text{-}1\} & \text{predRemoveFeature}(\text{pl},\ \text{pl2},\ s,\ \text{its},\ \text{pairs},\ P,\ Q) \\
\{\text{-}2\} & \textsc{forall}\ c\colon\ s(c) \Rightarrow \text{prod}(\text{pl},\ c) = \text{prod}(\text{pl2},\ c) \\
\hline
\{1\} & (s \subseteq \text{semantics}(F(\text{pl2})))
\end{array}
$$

Expanding the definition of predRemoveFeature, `removeFeaturePartRefStrong.2`:

$$
\begin{array}{l|ll}
& \{\text{-}1\} & (\text{syntaxRemoveFeature}(F(\text{pl}),\ F(\text{pl2}),\ A(\text{pl}),\ A(\text{pl2}),\ K(\text{pl}),\ K(\text{pl2}),\ P,\ Q,\ \text{its}, \\
& & \qquad\qquad\qquad\qquad \text{pairs}) \\
& & \quad \wedge \\
& & \quad \text{conditionsRemoveFeature}(F(\text{pl}),\ \text{its},\ \text{pairs},\ P,\ Q,\ K(\text{pl}))\ \wedge \\
& & \quad\ s\ =\ (\Diamond)(F(\text{pl}),\ \text{NOT\_FORMULA}(\text{NAME\_FORMULA}(Q)))) \\
& \{\text{-}2\} & \textsc{forall}\ c\colon\ s(c)\ \Rightarrow\ \text{prod}(\text{pl},\ c)\ =\ \text{prod}(\text{pl2},\ c) \\
\hline
& \{1\} & (s \subseteq \text{semantics}(F(\text{pl2})))
\end{array}
$$

Applying bddsimp,
`removeFeaturePartRefStrong.2`:

$$
\begin{array}{l|ll}
& \{\text{-}1\} & \text{syntaxRemoveFeature}(F(\text{pl}),\ F(\text{pl2}),\ A(\text{pl}),\ A(\text{pl2}),\ K(\text{pl}),\ K(\text{pl2}),\ P,\ Q,\ \text{its}, \\
& & \qquad\qquad\qquad\qquad \text{pairs}) \\
& \{\text{-}2\} & \text{conditionsRemoveFeature}(F(\text{pl}),\ \text{its},\ \text{pairs},\ P,\ Q,\ K(\text{pl})) \\
& \{\text{-}3\} & s\ =\ (\Diamond)(F(\text{pl}),\ \text{NOT\_FORMULA}(\text{NAME\_FORMULA}(Q))) \\
& \{\text{-}4\} & \textsc{forall}\ c\colon\ s(c)\ \Rightarrow\ \text{prod}(\text{pl},\ c)\ =\ \text{prod}(\text{pl2},\ c) \\
\hline
& \{1\} & (s \subseteq \text{semantics}(F(\text{pl2})))
\end{array}
$$

Expanding the definition of ¡¿,
`removeFeaturePartRefStrong.2`:

$$
\begin{array}{l|ll}
& \{\text{-}1\} & \text{syntaxRemoveFeature}(F(\text{pl}),\ F(\text{pl2}),\ A(\text{pl}),\ A(\text{pl2}),\ K(\text{pl}),\ K(\text{pl2}),\ P,\ Q,\ \text{its}, \\
& & \qquad\qquad\qquad\qquad \text{pairs}) \\
& \{\text{-}2\} & \text{conditionsRemoveFeature}(F(\text{pl}),\ \text{its},\ \text{pairs},\ P,\ Q,\ K(\text{pl})) \\
& \{\text{-}3\} & s\ =\ (\{c\ \mid\ \text{semantics}(F(\text{pl}))(c)\ \wedge\ \text{satisfies}(\text{NOT\_FORMULA}(\text{NAME\_FORMULA}(Q)), \\
& \{\text{-}4\} & \textsc{forall}\ c\colon\ s(c)\ \Rightarrow\ \text{prod}(\text{pl},\ c)\ =\ \text{prod}(\text{pl2},\ c) \\
\hline
& \{1\} & (s \subseteq \text{semantics}(F(\text{pl2})))
\end{array}
$$

Applying decompose-equality,
`removeFeaturePartRefStrong.2`:

$$
\begin{array}{l|ll}
& \{\text{-}1\} & \forall\ (x\colon\ \text{Configuration})\colon \\
& & \quad s(x)\ =\ (\text{semantics}(F(\text{pl}))(x)\ \wedge\ \text{satisfies}(\text{NOT\_FORMULA}(\text{NAME\_FORMULA}(Q)), \\
& \{\text{-}2\} & \text{syntaxRemoveFeature}(F(\text{pl}),\ F(\text{pl2}),\ A(\text{pl}),\ A(\text{pl2}),\ K(\text{pl}),\ K(\text{pl2}),\ P,\ Q,\ \text{its}, \\
& & \qquad\qquad\qquad\qquad \text{pairs}) \\
& \{\text{-}3\} & \text{conditionsRemoveFeature}(F(\text{pl}),\ \text{its},\ \text{pairs},\ P,\ Q,\ K(\text{pl})) \\
& \{\text{-}4\} & \textsc{forall}\ c\colon\ s(c)\ \Rightarrow\ \text{prod}(\text{pl},\ c)\ =\ \text{prod}(\text{pl2},\ c) \\
\hline
& \{1\} & (s \subseteq \text{semantics}(F(\text{pl2})))
\end{array}
$$

Expanding the definition of subset?,

`removeFeaturePartRefStrong.2:`

$$
\begin{array}{l|ll}
& \{-1\} & \forall\ (x\colon \text{Configuration})\colon \\
& & \quad s(x) = (\text{semantics}(F(\text{pl}))(x) \wedge \text{satisfies}(\text{NOT\_FORMULA}(\text{NAME\_FORMULA}(Q)), \\
& \{-2\} & \text{syntaxRemoveFeature}(F(\text{pl}), F(\text{pl2}), A(\text{pl}), A(\text{pl2}), K(\text{pl}), K(\text{pl2}), P, Q, \text{its}, \\
& & \qquad \text{pairs}) \\
& \{-3\} & \text{conditionsRemoveFeature}(F(\text{pl}), \text{its}, \text{pairs}, P, Q, K(\text{pl})) \\
& \{-4\} & \text{FORALL } c\colon s(c) \Rightarrow \text{prod}(\text{pl}, c) = \text{prod}(\text{pl2}, c) \\
\hline
& \{1\} & \text{FORALL } (x\colon \text{Configuration})\colon (x \in s) \Rightarrow (x \in \text{semantics}(F(\text{pl2})))
\end{array}
$$

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

$$
\begin{array}{l|ll}
& \{-1\} & \forall\ (x\colon \text{Configuration})\colon \\
& & \quad s(x) = (\text{semantics}(F(\text{pl}))(x) \wedge \text{satisfies}(\text{NOT\_FORMULA}(\text{NAME\_FORMULA}(Q)), \\
& \{-2\} & \text{syntaxRemoveFeature}(F(\text{pl}), F(\text{pl2}), A(\text{pl}), A(\text{pl2}), K(\text{pl}), K(\text{pl2}), P, Q, \text{its}, \\
& & \qquad \text{pairs}) \\
& \{-3\} & \text{conditionsRemoveFeature}(F(\text{pl}), \text{its}, \text{pairs}, P, Q, K(\text{pl})) \\
& \{-4\} & \text{FORALL } c\colon s(c) \Rightarrow \text{prod}(\text{pl}, c) = \text{prod}(\text{pl2}, c) \\
\hline
& \{1\} & (c \in s) \Rightarrow (c \in \text{semantics}(F(\text{pl2})))
\end{array}
$$

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

$$
\begin{array}{l|ll}
& \{-1\} & s(c) = (\text{semantics}(F(\text{pl}))(c) \wedge \text{satisfies}(\text{NOT\_FORMULA}(\text{NAME\_FORMULA}(Q)), c \\
& \{-2\} & \text{syntaxRemoveFeature}(F(\text{pl}), F(\text{pl2}), A(\text{pl}), A(\text{pl2}), K(\text{pl}), K(\text{pl2}), P, Q, \text{its}, \\
& & \qquad \text{pairs}) \\
& \{-3\} & \text{conditionsRemoveFeature}(F(\text{pl}), \text{its}, \text{pairs}, P, Q, K(\text{pl})) \\
& \{-4\} & \text{FORALL } c\colon s(c) \Rightarrow \text{prod}(\text{pl}, c) = \text{prod}(\text{pl2}, c) \\
\hline
& \{1\} & (c \in s) \Rightarrow (c \in \text{semantics}(F(\text{pl2})))
\end{array}
$$

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

$$
\begin{array}{l|ll}
& \{-1\} & s(c) = (\text{semantics}(F(\text{pl}))(c) \wedge \text{satisfies}(\text{NOT\_FORMULA}(\text{NAME\_FORMULA}(Q)), c \\
& \{-2\} & \text{syntaxRemoveFeature}(F(\text{pl}), F(\text{pl2}), A(\text{pl}), A(\text{pl2}), K(\text{pl}), K(\text{pl2}), P, Q, \text{its}, \\
& & \qquad \text{pairs}) \\
& \{-3\} & \text{conditionsRemoveFeature}(F(\text{pl}), \text{its}, \text{pairs}, P, Q, K(\text{pl})) \\
& \{-4\} & \text{FORALL } c\colon s(c) \Rightarrow \text{prod}(\text{pl}, c) = \text{prod}(\text{pl2}, c) \\
\hline
& \{1\} & s(c) \Rightarrow \text{semantics}(F(\text{pl2}))(c)
\end{array}
$$

Expanding the definition of syntaxRemoveFeature,

`removeFeaturePartRefStrong.2:`

{-1}	$s(c) = (\text{semantics}(F(\text{pl}))(c) \wedge \text{satisfies}(\text{NOT\_FORMULA}(\text{NAME\_FORMULA}(Q)),\ c$
{-2}	$\text{removeFeature}(F(\text{pl}),\ F(\text{pl2}),\ P,\ Q) \wedge$ $\text{features}(F(\text{pl}))(P) \wedge$ $\text{features}(F(\text{pl}))(Q) \wedge$ $A(\text{pl}) = \text{overw}(\text{pairs},\ A(\text{pl2})) \wedge K(\text{pl2}) = (K(\text{pl}) \setminus \text{its})$
{-3}	$\text{conditionsRemoveFeature}(F(\text{pl}),\ \text{its},\ \text{pairs},\ P,\ Q,\ K(\text{pl}))$
{-4}	$\text{FORALL } c\text{: } s(c) \Rightarrow \text{prod}(\text{pl},\ c) = \text{prod}(\text{pl2},\ c)$
{1}	$s(c) \Rightarrow \text{semantics}(F(\text{pl2}))(c)$

Applying bddsimp,
`removeFeaturePartRefStrong.2:`

{-1}	$s(c)$
{-2}	$\text{semantics}(F(\text{pl}))(c)$
{-3}	$\text{satisfies}(\text{NOT\_FORMULA}(\text{NAME\_FORMULA}(Q)),\ c)$
{-4}	$\text{removeFeature}(F(\text{pl}),\ F(\text{pl2}),\ P,\ Q)$
{-5}	$\text{features}(F(\text{pl}))(P)$
{-6}	$\text{features}(F(\text{pl}))(Q)$
{-7}	$A(\text{pl}) = \text{overw}(\text{pairs},\ A(\text{pl2}))$
{-8}	$K(\text{pl2}) = (K(\text{pl}) \setminus \text{its})$
{-9}	$\text{conditionsRemoveFeature}(F(\text{pl}),\ \text{its},\ \text{pairs},\ P,\ Q,\ K(\text{pl}))$
{-10}	$\text{FORALL } c\text{: } s(c) \Rightarrow \text{prod}(\text{pl},\ c) = \text{prod}(\text{pl2},\ c)$
{1}	$\text{semantics}(F(\text{pl2}))(c)$

Expanding the definition of semantics,
`removeFeaturePartRefStrong.2:`

{-1}	$s(c)$
{-2}	$\text{satImpConsts}(F(\text{pl}),\ c) \wedge \text{satExpConsts}(F(\text{pl}),\ c)$
{-3}	$\text{satisfies}(\text{NOT\_FORMULA}(\text{NAME\_FORMULA}(Q)),\ c)$
{-4}	$\text{removeFeature}(F(\text{pl}),\ F(\text{pl2}),\ P,\ Q)$
{-5}	$\text{features}(F(\text{pl}))(P)$
{-6}	$\text{features}(F(\text{pl}))(Q)$
{-7}	$A(\text{pl}) = \text{overw}(\text{pairs},\ A(\text{pl2}))$
{-8}	$K(\text{pl2}) = (K(\text{pl}) \setminus \text{its})$
{-9}	$\text{conditionsRemoveFeature}(F(\text{pl}),\ \text{its},\ \text{pairs},\ P,\ Q,\ K(\text{pl}))$
{-10}	$\text{FORALL } c\text{: } s(c) \Rightarrow \text{prod}(\text{pl},\ c) = \text{prod}(\text{pl2},\ c)$
{1}	$\text{satImpConsts}(F(\text{pl2}),\ c) \wedge \text{satExpConsts}(F(\text{pl2}),\ c)$

Applying bddsimp,
we get 2 subgoals:

`removeFeaturePartRefStrong.2.1:`

$$
\begin{array}{c|l}
\{\text{-}1\} & s(c) \\
\{\text{-}2\} & \text{satImpConsts}(F(\text{pl}),\ c) \\
\{\text{-}3\} & \text{satExpConsts}(F(\text{pl}),\ c) \\
\{\text{-}4\} & \text{satisfies}(\text{NOT\_FORMULA}(\text{NAME\_FORMULA}(Q)),\ c) \\
\{\text{-}5\} & \text{removeFeature}(F(\text{pl}),\ F(\text{pl2}),\ P,\ Q) \\
\{\text{-}6\} & \text{features}(F(\text{pl}))(P) \\
\{\text{-}7\} & \text{features}(F(\text{pl}))(Q) \\
\{\text{-}8\} & A(\text{pl}) = \text{overw}(\text{pairs},\ A(\text{pl2})) \\
\{\text{-}9\} & K(\text{pl2}) = (K(\text{pl}) \setminus \text{its}) \\
\{\text{-}10\} & \text{conditionsRemoveFeature}(F(\text{pl}),\ \text{its},\ \text{pairs},\ P,\ Q,\ K(\text{pl})) \\
\{\text{-}11\} & \textsc{forall}\ c\colon\ s(c) \Rightarrow \text{prod}(\text{pl},\ c) = \text{prod}(\text{pl2},\ c) \\
\hline
\{1\} & \text{satExpConsts}(F(\text{pl2}),\ c)
\end{array}
$$

Expanding the definition of satExpConsts,
`removeFeaturePartRefStrong.2.1:`

$$
\begin{array}{c|l}
\{\text{-}1\} & s(c) \\
\{\text{-}2\} & \text{satImpConsts}(F(\text{pl}),\ c) \\
\{\text{-}3\} & \textsc{forall}\ (f\colon\ \text{Formula\_})\colon\ \text{formulae}(F(\text{pl}))(f) \Rightarrow \text{satisfies}(f,\ c) \\
\{\text{-}4\} & \text{satisfies}(\text{NOT\_FORMULA}(\text{NAME\_FORMULA}(Q)),\ c) \\
\{\text{-}5\} & \text{removeFeature}(F(\text{pl}),\ F(\text{pl2}),\ P,\ Q) \\
\{\text{-}6\} & \text{features}(F(\text{pl}))(P) \\
\{\text{-}7\} & \text{features}(F(\text{pl}))(Q) \\
\{\text{-}8\} & A(\text{pl}) = \text{overw}(\text{pairs},\ A(\text{pl2})) \\
\{\text{-}9\} & K(\text{pl2}) = (K(\text{pl}) \setminus \text{its}) \\
\{\text{-}10\} & \text{conditionsRemoveFeature}(F(\text{pl}),\ \text{its},\ \text{pairs},\ P,\ Q,\ K(\text{pl})) \\
\{\text{-}11\} & \textsc{forall}\ c\colon\ s(c) \Rightarrow \text{prod}(\text{pl},\ c) = \text{prod}(\text{pl2},\ c) \\
\hline
\{1\} & \textsc{forall}\ (f\colon\ \text{Formula\_})\colon\ \text{formulae}(F(\text{pl2}))(f) \Rightarrow \text{satisfies}(f,\ c)
\end{array}
$$

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

`removeFeaturePartRefStrong.2.1:`

{-1}	$s(c)$
{-2}	satImpConsts(F(pl), c)
{-3}	FORALL (f: Formula_): formulae(F(pl))(f) $\Rightarrow$ satisfies(f, c)
{-4}	satisfies(NOT_FORMULA(NAME_FORMULA(Q)), c)
{-5}	removeFeature(F(pl), F(pl2), P, Q)
{-6}	features(F(pl))(P)
{-7}	features(F(pl))(Q)
{-8}	A(pl) = overw(pairs, A(pl2))
{-9}	K(pl2) = (K(pl) $\setminus$ its)
{-10}	conditionsRemoveFeature(F(pl), its, pairs, P, Q, K(pl))
{-11}	FORALL c: $s(c)$ $\Rightarrow$ prod(pl, c) = prod(pl2, c)
{1}	formulae(F(pl2))(f) $\Rightarrow$ satisfies(f, c)

Instantiating the top quantifier in -3 with the terms: f,
`removeFeaturePartRefStrong.2.1:`

{-1}	$s(c)$
{-2}	satImpConsts(F(pl), c)
{-3}	formulae(F(pl))(f) $\Rightarrow$ satisfies(f, c)
{-4}	satisfies(NOT_FORMULA(NAME_FORMULA(Q)), c)
{-5}	removeFeature(F(pl), F(pl2), P, Q)
{-6}	features(F(pl))(P)
{-7}	features(F(pl))(Q)
{-8}	A(pl) = overw(pairs, A(pl2))
{-9}	K(pl2) = (K(pl) $\setminus$ its)
{-10}	conditionsRemoveFeature(F(pl), its, pairs, P, Q, K(pl))
{-11}	FORALL c: $s(c)$ $\Rightarrow$ prod(pl, c) = prod(pl2, c)
{1}	formulae(F(pl2))(f) $\Rightarrow$ satisfies(f, c)

Expanding the definition of removeFeature,

`removeFeaturePartRefStrong.2.1:`

$$
\begin{array}{c|l}
\{-1\} & s(c) \\
\{-2\} & \text{satImpConsts}(F(\text{pl}),\ c) \\
\{-3\} & \text{formulae}(F(\text{pl}))(f) \Rightarrow \text{satisfies}(f,\ c) \\
\{-4\} & \text{satisfies}(\text{NOT\_FORMULA}(\text{NAME\_FORMULA}(Q)),\ c) \\
\{-5\} & \text{formulae}(F(\text{pl2})) = \text{filterFormulae}(F(\text{pl}),\ Q) \wedge \\
 & \quad \text{features}(F(\text{pl2})) = (\text{features}(F(\text{pl})) \setminus \{Q\}) \\
\{-6\} & \text{features}(F(\text{pl}))(P) \\
\{-7\} & \text{features}(F(\text{pl}))(Q) \\
\{-8\} & A(\text{pl}) = \text{overw}(\text{pairs},\ A(\text{pl2})) \\
\{-9\} & K(\text{pl2}) = (K(\text{pl}) \setminus \text{its}) \\
\{-10\} & \text{conditionsRemoveFeature}(F(\text{pl}),\ \text{its},\ \text{pairs},\ P,\ Q,\ K(\text{pl})) \\
\{-11\} & \textsc{forall}\ c\colon\ s(c) \Rightarrow \text{prod}(\text{pl},\ c) = \text{prod}(\text{pl2},\ c) \\
\hline
\{1\} & \text{formulae}(F(\text{pl2}))(f) \Rightarrow \text{satisfies}(f,\ c)
\end{array}
$$

Expanding the definition of filterFormulae,
`removeFeaturePartRefStrong.2.1:`

$$
\begin{array}{c|l}
\{-1\} & s(c) \\
\{-2\} & \text{satImpConsts}(F(\text{pl}),\ c) \\
\{-3\} & \text{formulae}(F(\text{pl}))(f) \Rightarrow \text{satisfies}(f,\ c) \\
\{-4\} & \text{satisfies}(\text{NOT\_FORMULA}(\text{NAME\_FORMULA}(Q)),\ c) \\
\{-5\} & (\text{formulae}(F(\text{pl2})) = \\
 & \qquad (\{\text{form}\colon\ \text{Formula\_} \mid \text{formulae}(F(\text{pl}))(\text{form}) \wedge \neg\ (Q \in \text{names}(\text{form}))\})) \\
 & \quad \wedge\ \text{features}(F(\text{pl2})) = (\text{features}(F(\text{pl})) \setminus \{Q\}) \\
\{-6\} & \text{features}(F(\text{pl}))(P) \\
\{-7\} & \text{features}(F(\text{pl}))(Q) \\
\{-8\} & A(\text{pl}) = \text{overw}(\text{pairs},\ A(\text{pl2})) \\
\{-9\} & K(\text{pl2}) = (K(\text{pl}) \setminus \text{its}) \\
\{-10\} & \text{conditionsRemoveFeature}(F(\text{pl}),\ \text{its},\ \text{pairs},\ P,\ Q,\ K(\text{pl})) \\
\{-11\} & \textsc{forall}\ c\colon\ s(c) \Rightarrow \text{prod}(\text{pl},\ c) = \text{prod}(\text{pl2},\ c) \\
\hline
\{1\} & \text{formulae}(F(\text{pl2}))(f) \Rightarrow \text{satisfies}(f,\ c)
\end{array}
$$

Applying disjunctive simplification to flatten sequent,

removeFeaturePartRefStrong.2.1:

$$\begin{array}{r|l}
\{-1\} & s(c) \\
\{-2\} & \text{satImpConsts}(F(\text{pl}),\ c) \\
\{-3\} & \text{formulae}(F(\text{pl}))(f) \ \Rightarrow\ \text{satisfies}(f,\ c) \\
\{-4\} & \text{satisfies}(\text{NOT_FORMULA}(\text{NAME_FORMULA}(Q)),\ c) \\
\{-5\} & \text{formulae}(F(\text{pl2})) \ = \\
 & \quad (\{\text{form}:\ \text{Formula_} \mid \text{formulae}(F(\text{pl}))(\text{form}) \ \wedge\ \neg\ (Q \in \text{names}(\text{form}))\}) \\
\{-6\} & \text{features}(F(\text{pl2})) \ = \ (\text{features}(F(\text{pl})) \setminus \{Q\}) \\
\{-7\} & \text{features}(F(\text{pl}))(P) \\
\{-8\} & \text{features}(F(\text{pl}))(Q) \\
\{-9\} & A(\text{pl}) \ = \ \text{overw}(\text{pairs},\ A(\text{pl2})) \\
\{-10\} & K(\text{pl2}) \ = \ (K(\text{pl}) \setminus \text{its}) \\
\{-11\} & \text{conditionsRemoveFeature}(F(\text{pl}),\ \text{its},\ \text{pairs},\ P,\ Q,\ K(\text{pl})) \\
\{-12\} & \text{FORALL}\ c:\ s(c) \ \Rightarrow\ \text{prod}(\text{pl},\ c) \ = \ \text{prod}(\text{pl2},\ c) \\
\{-13\} & \text{formulae}(F(\text{pl2}))(f) \\
\hline
\{1\} & \text{satisfies}(f,\ c)
\end{array}$$

Applying decompose-equality,

removeFeaturePartRefStrong.2.1:

$$\begin{array}{r|l}
\{-1\} & \forall\ (x:\ \text{Formula_}): \\
 & \quad \text{formulae}(F(\text{pl2}))(x) \ = \ (\text{formulae}(F(\text{pl}))(x) \ \wedge\ \neg\ (Q \in \text{names}(x))) \\
\{-2\} & s(c) \\
\{-3\} & \text{satImpConsts}(F(\text{pl}),\ c) \\
\{-4\} & \text{formulae}(F(\text{pl}))(f) \ \Rightarrow\ \text{satisfies}(f,\ c) \\
\{-5\} & \text{satisfies}(\text{NOT_FORMULA}(\text{NAME_FORMULA}(Q)),\ c) \\
\{-6\} & \text{features}(F(\text{pl2})) \ = \ (\text{features}(F(\text{pl})) \setminus \{Q\}) \\
\{-7\} & \text{features}(F(\text{pl}))(P) \\
\{-8\} & \text{features}(F(\text{pl}))(Q) \\
\{-9\} & A(\text{pl}) \ = \ \text{overw}(\text{pairs},\ A(\text{pl2})) \\
\{-10\} & K(\text{pl2}) \ = \ (K(\text{pl}) \setminus \text{its}) \\
\{-11\} & \text{conditionsRemoveFeature}(F(\text{pl}),\ \text{its},\ \text{pairs},\ P,\ Q,\ K(\text{pl})) \\
\{-12\} & \text{FORALL}\ c:\ s(c) \ \Rightarrow\ \text{prod}(\text{pl},\ c) \ = \ \text{prod}(\text{pl2},\ c) \\
\{-13\} & \text{formulae}(F(\text{pl2}))(f) \\
\hline
\{1\} & \text{satisfies}(f,\ c)
\end{array}$$

Instantiating the top quantifier in -1 with the terms: f, `removeFeaturePartRefStrong.2.1`:

{-1}	formulae(F(pl2))(f) = (formulae(F(pl))(f) $\wedge$ $\neg$ ($Q \in$ names(f)))
{-2}	$s(c)$
{-3}	satImpConsts(F(pl), c)
{-4}	formulae(F(pl))(f) $\Rightarrow$ satisfies(f, c)
{-5}	satisfies(NOT_FORMULA(NAME_FORMULA(Q)), c)
{-6}	features(F(pl2)) = (features(F(pl)) $\setminus$ {Q})
{-7}	features(F(pl))(P)
{-8}	features(F(pl))(Q)
{-9}	A(pl) = overw(pairs, A(pl2))
{-10}	K(pl2) = (K(pl) $\setminus$ its)
{-11}	conditionsRemoveFeature(F(pl), its, pairs, P, Q, K(pl))
{-12}	FORALL c: $s(c)$ $\Rightarrow$ prod(pl, c) = prod(pl2, c)
{-13}	formulae(F(pl2))(f)
{1}	satisfies(f, c)

Applying bddsimp,
This completes the proof of `removeFeaturePartRefStrong.2.1`.
`removeFeaturePartRefStrong.2.2`:

{-1}	$s(c)$
{-2}	satImpConsts(F(pl), c)
{-3}	satExpConsts(F(pl), c)
{-4}	satisfies(NOT_FORMULA(NAME_FORMULA(Q)), c)
{-5}	removeFeature(F(pl), F(pl2), P, Q)
{-6}	features(F(pl))(P)
{-7}	features(F(pl))(Q)
{-8}	A(pl) = overw(pairs, A(pl2))
{-9}	K(pl2) = (K(pl) $\setminus$ its)
{-10}	conditionsRemoveFeature(F(pl), its, pairs, P, Q, K(pl))
{-11}	FORALL c: $s(c)$ $\Rightarrow$ prod(pl, c) = prod(pl2, c)
{1}	satImpConsts(F(pl2), c)

Expanding the definition of satImpConsts,

`removeFeaturePartRefStrong.2.2:`

{-1}	$s(c)$
{-2}	FORALL $(n\colon$ Name$)\colon\ c(n) \Rightarrow$ features$(F(\mathrm{pl}))(n)$
{-3}	satExpConsts$(F(\mathrm{pl}),\ c)$
{-4}	satisfies(NOT_FORMULA(NAME_FORMULA(Q))$,\ c)$
{-5}	removeFeature$(F(\mathrm{pl}),\ F(\mathrm{pl2}),\ P,\ Q)$
{-6}	features$(F(\mathrm{pl}))(P)$
{-7}	features$(F(\mathrm{pl}))(Q)$
{-8}	$A(\mathrm{pl}) = \mathrm{overw}(\mathrm{pairs},\ A(\mathrm{pl2}))$
{-9}	$K(\mathrm{pl2}) = (K(\mathrm{pl}) \setminus \mathrm{its})$
{-10}	conditionsRemoveFeature$(F(\mathrm{pl}),\ \mathrm{its},\ \mathrm{pairs},\ P,\ Q,\ K(\mathrm{pl}))$
{-11}	FORALL $c\colon\ s(c) \Rightarrow \mathrm{prod}(\mathrm{pl},\ c) = \mathrm{prod}(\mathrm{pl2},\ c)$
{1}	FORALL $(n\colon$ Name$)\colon\ c(n) \Rightarrow$ features$(F(\mathrm{pl2}))(n)$

For the top quantifier in 1, we introduce Skolem constants: n,
`removeFeaturePartRefStrong.2.2:`

{-1}	$s(c)$
{-2}	FORALL $(n\colon$ Name$)\colon\ c(n) \Rightarrow$ features$(F(\mathrm{pl}))(n)$
{-3}	satExpConsts$(F(\mathrm{pl}),\ c)$
{-4}	satisfies(NOT_FORMULA(NAME_FORMULA(Q))$,\ c)$
{-5}	removeFeature$(F(\mathrm{pl}),\ F(\mathrm{pl2}),\ P,\ Q)$
{-6}	features$(F(\mathrm{pl}))(P)$
{-7}	features$(F(\mathrm{pl}))(Q)$
{-8}	$A(\mathrm{pl}) = \mathrm{overw}(\mathrm{pairs},\ A(\mathrm{pl2}))$
{-9}	$K(\mathrm{pl2}) = (K(\mathrm{pl}) \setminus \mathrm{its})$
{-10}	conditionsRemoveFeature$(F(\mathrm{pl}),\ \mathrm{its},\ \mathrm{pairs},\ P,\ Q,\ K(\mathrm{pl}))$
{-11}	FORALL $c\colon\ s(c) \Rightarrow \mathrm{prod}(\mathrm{pl},\ c) = \mathrm{prod}(\mathrm{pl2},\ c)$
{1}	$c(n) \Rightarrow$ features$(F(\mathrm{pl2}))(n)$

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

`removeFeaturePartRefStrong.2.2:`

{-1}	$s(c)$
{-2}	$c(n) \Rightarrow$ features$(F(\text{pl}))(n)$
{-3}	satExpConsts$(F(\text{pl}),\ c)$
{-4}	satisfies(NOT_FORMULA(NAME_FORMULA(Q)), c)
{-5}	removeFeature$(F(\text{pl}),\ F(\text{pl2}),\ P,\ Q)$
{-6}	features$(F(\text{pl}))(P)$
{-7}	features$(F(\text{pl}))(Q)$
{-8}	$A(\text{pl}) = \text{overw}(\text{pairs},\ A(\text{pl2}))$
{-9}	$K(\text{pl2}) = (K(\text{pl}) \setminus \text{its})$
{-10}	conditionsRemoveFeature$(F(\text{pl}),\ \text{its},\ \text{pairs},\ P,\ Q,\ K(\text{pl}))$
{-11}	FORALL c: $s(c) \Rightarrow \text{prod}(\text{pl},\ c) = \text{prod}(\text{pl2},\ c)$
{1}	$c(n) \Rightarrow$ features$(F(\text{pl2}))(n)$

Applying bddsimp,

`removeFeaturePartRefStrong.2.2:`

{-1}	$s(c)$
{-2}	$c(n)$
{-3}	features$(F(\text{pl}))(n)$
{-4}	satExpConsts$(F(\text{pl}),\ c)$
{-5}	satisfies(NOT_FORMULA(NAME_FORMULA(Q)), c)
{-6}	removeFeature$(F(\text{pl}),\ F(\text{pl2}),\ P,\ Q)$
{-7}	features$(F(\text{pl}))(P)$
{-8}	features$(F(\text{pl}))(Q)$
{-9}	$A(\text{pl}) = \text{overw}(\text{pairs},\ A(\text{pl2}))$
{-10}	$K(\text{pl2}) = (K(\text{pl}) \setminus \text{its})$
{-11}	conditionsRemoveFeature$(F(\text{pl}),\ \text{its},\ \text{pairs},\ P,\ Q,\ K(\text{pl}))$
{-12}	FORALL c: $s(c) \Rightarrow \text{prod}(\text{pl},\ c) = \text{prod}(\text{pl2},\ c)$
{1}	features$(F(\text{pl2}))(n)$

Expanding the definition of removeFeature,

`removeFeaturePartRefStrong.2.2:`

$$
\begin{array}{r|l}
\{\text{-}1\} & s(c) \\
\{\text{-}2\} & c(n) \\
\{\text{-}3\} & \mathrm{features}(F(\mathrm{pl}))(n) \\
\{\text{-}4\} & \mathrm{satExpConsts}(F(\mathrm{pl}),\ c) \\
\{\text{-}5\} & \mathrm{satisfies}(\mathrm{NOT\_FORMULA}(\mathrm{NAME\_FORMULA}(Q)),\ c) \\
\{\text{-}6\} & \mathrm{formulae}(F(\mathrm{pl2}))\ =\ \mathrm{filterFormulae}(F(\mathrm{pl}),\ Q)\ \wedge \\
 & \quad \mathrm{features}(F(\mathrm{pl2}))\ =\ (\mathrm{features}(F(\mathrm{pl})) \setminus \{Q\}) \\
\{\text{-}7\} & \mathrm{features}(F(\mathrm{pl}))(P) \\
\{\text{-}8\} & \mathrm{features}(F(\mathrm{pl}))(Q) \\
\{\text{-}9\} & A(\mathrm{pl})\ =\ \mathrm{overw}(\mathrm{pairs},\ A(\mathrm{pl2})) \\
\{\text{-}10\} & K(\mathrm{pl2})\ =\ (K(\mathrm{pl}) \setminus \mathrm{its}) \\
\{\text{-}11\} & \mathrm{conditionsRemoveFeature}(F(\mathrm{pl}),\ \mathrm{its},\ \mathrm{pairs},\ P,\ Q,\ K(\mathrm{pl})) \\
\{\text{-}12\} & \textsc{forall}\ c\colon\ s(c)\ \Rightarrow\ \mathrm{prod}(\mathrm{pl},\ c)\ =\ \mathrm{prod}(\mathrm{pl2},\ c) \\
\hline
\{1\} & \mathrm{features}(F(\mathrm{pl2}))(n)
\end{array}
$$

Applying disjunctive simplification to flatten sequent,
`removeFeaturePartRefStrong.2.2:`

$$
\begin{array}{r|l}
\{\text{-}1\} & s(c) \\
\{\text{-}2\} & c(n) \\
\{\text{-}3\} & \mathrm{features}(F(\mathrm{pl}))(n) \\
\{\text{-}4\} & \mathrm{satExpConsts}(F(\mathrm{pl}),\ c) \\
\{\text{-}5\} & \mathrm{satisfies}(\mathrm{NOT\_FORMULA}(\mathrm{NAME\_FORMULA}(Q)),\ c) \\
\{\text{-}6\} & \mathrm{formulae}(F(\mathrm{pl2}))\ =\ \mathrm{filterFormulae}(F(\mathrm{pl}),\ Q) \\
\{\text{-}7\} & \mathrm{features}(F(\mathrm{pl2}))\ =\ (\mathrm{features}(F(\mathrm{pl})) \setminus \{Q\}) \\
\{\text{-}8\} & \mathrm{features}(F(\mathrm{pl}))(P) \\
\{\text{-}9\} & \mathrm{features}(F(\mathrm{pl}))(Q) \\
\{\text{-}10\} & A(\mathrm{pl})\ =\ \mathrm{overw}(\mathrm{pairs},\ A(\mathrm{pl2})) \\
\{\text{-}11\} & K(\mathrm{pl2})\ =\ (K(\mathrm{pl}) \setminus \mathrm{its}) \\
\{\text{-}12\} & \mathrm{conditionsRemoveFeature}(F(\mathrm{pl}),\ \mathrm{its},\ \mathrm{pairs},\ P,\ Q,\ K(\mathrm{pl})) \\
\{\text{-}13\} & \textsc{forall}\ c\colon\ s(c)\ \Rightarrow\ \mathrm{prod}(\mathrm{pl},\ c)\ =\ \mathrm{prod}(\mathrm{pl2},\ c) \\
\hline
\{1\} & \mathrm{features}(F(\mathrm{pl2}))(n)
\end{array}
$$

Expanding the definition of remove,

`removeFeaturePartRefStrong.2.2:`

{-1}	$s(c)$
{-2}	$c(n)$
{-3}	$\text{features}(F(\text{pl}))(n)$
{-4}	$\text{satExpConsts}(F(\text{pl}),\ c)$
{-5}	$\text{satisfies}(\text{NOT\_FORMULA}(\text{NAME\_FORMULA}(Q)),\ c)$
{-6}	$\text{formulae}(F(\text{pl2})) = \text{filterFormulae}(F(\text{pl}),\ Q)$
{-7}	$\text{features}(F(\text{pl2})) = (\{y \mid Q \neq y \wedge (y \in \text{features}(F(\text{pl})))\})$
{-8}	$\text{features}(F(\text{pl}))(P)$
{-9}	$\text{features}(F(\text{pl}))(Q)$
{-10}	$A(\text{pl}) = \text{overw}(\text{pairs},\ A(\text{pl2}))$
{-11}	$K(\text{pl2}) = (K(\text{pl}) \setminus \text{its})$
{-12}	$\text{conditionsRemoveFeature}(F(\text{pl}),\ \text{its},\ \text{pairs},\ P,\ Q,\ K(\text{pl}))$
{-13}	FORALL $c\colon s(c) \Rightarrow \text{prod}(\text{pl},\ c) = \text{prod}(\text{pl2},\ c)$
{1}	$\text{features}(F(\text{pl2}))(n)$

Applying decompose-equality,
`removeFeaturePartRefStrong.2.2:`

{-1}	$\forall\ (x\colon \text{Name})\colon \text{features}(F(\text{pl2}))(x) = (Q \neq x \wedge (x \in \text{features}(F(\text{pl}))))$
{-2}	$s(c)$
{-3}	$c(n)$
{-4}	$\text{features}(F(\text{pl}))(n)$
{-5}	$\text{satExpConsts}(F(\text{pl}),\ c)$
{-6}	$\text{satisfies}(\text{NOT\_FORMULA}(\text{NAME\_FORMULA}(Q)),\ c)$
{-7}	$\text{formulae}(F(\text{pl2})) = \text{filterFormulae}(F(\text{pl}),\ Q)$
{-8}	$\text{features}(F(\text{pl}))(P)$
{-9}	$\text{features}(F(\text{pl}))(Q)$
{-10}	$A(\text{pl}) = \text{overw}(\text{pairs},\ A(\text{pl2}))$
{-11}	$K(\text{pl2}) = (K(\text{pl}) \setminus \text{its})$
{-12}	$\text{conditionsRemoveFeature}(F(\text{pl}),\ \text{its},\ \text{pairs},\ P,\ Q,\ K(\text{pl}))$
{-13}	FORALL $c\colon s(c) \Rightarrow \text{prod}(\text{pl},\ c) = \text{prod}(\text{pl2},\ c)$
{1}	$\text{features}(F(\text{pl2}))(n)$

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

`removeFeaturePartRefStrong.2.2:`

{-1}	$\text{features}(F(\text{pl2}))(n) \ = \ (Q \ \neq \ n \ \wedge \ (n \in \text{features}(F(\text{pl}))))$
{-2}	$s(c)$
{-3}	$c(n)$
{-4}	$\text{features}(F(\text{pl}))(n)$
{-5}	$\text{satExpConsts}(F(\text{pl}),\ c)$
{-6}	$\text{satisfies}(\text{NOT\_FORMULA}(\text{NAME\_FORMULA}(Q)),\ c)$
{-7}	$\text{formulae}(F(\text{pl2})) \ = \ \text{filterFormulae}(F(\text{pl}),\ Q)$
{-8}	$\text{features}(F(\text{pl}))(P)$
{-9}	$\text{features}(F(\text{pl}))(Q)$
{-10}	$A(\text{pl}) \ = \ \text{overw}(\text{pairs},\ A(\text{pl2}))$
{-11}	$K(\text{pl2}) \ = \ (K(\text{pl}) \setminus \text{its})$
{-12}	$\text{conditionsRemoveFeature}(F(\text{pl}),\ \text{its},\ \text{pairs},\ P,\ Q,\ K(\text{pl}))$
{-13}	$\text{FORALL}\ c\colon\ s(c) \ \Rightarrow \ \text{prod}(\text{pl},\ c) \ = \ \text{prod}(\text{pl2},\ c)$
{1}	$\text{features}(F(\text{pl2}))(n)$

Expanding the definition of member,
`removeFeaturePartRefStrong.2.2:`

{-1}	$\text{features}(F(\text{pl2}))(n) \ = \ (Q \ \neq \ n \ \wedge \ \text{features}(F(\text{pl}))(n))$
{-2}	$s(c)$
{-3}	$c(n)$
{-4}	$\text{features}(F(\text{pl}))(n)$
{-5}	$\text{satExpConsts}(F(\text{pl}),\ c)$
{-6}	$\text{satisfies}(\text{NOT\_FORMULA}(\text{NAME\_FORMULA}(Q)),\ c)$
{-7}	$\text{formulae}(F(\text{pl2})) \ = \ \text{filterFormulae}(F(\text{pl}),\ Q)$
{-8}	$\text{features}(F(\text{pl}))(P)$
{-9}	$\text{features}(F(\text{pl}))(Q)$
{-10}	$A(\text{pl}) \ = \ \text{overw}(\text{pairs},\ A(\text{pl2}))$
{-11}	$K(\text{pl2}) \ = \ (K(\text{pl}) \setminus \text{its})$
{-12}	$\text{conditionsRemoveFeature}(F(\text{pl}),\ \text{its},\ \text{pairs},\ P,\ Q,\ K(\text{pl}))$
{-13}	$\text{FORALL}\ c\colon\ s(c) \ \Rightarrow \ \text{prod}(\text{pl},\ c) \ = \ \text{prod}(\text{pl2},\ c)$
{1}	$\text{features}(F(\text{pl2}))(n)$

Applying bddsimp,

`removeFeaturePartRefStrong.2.2:`

$$
\begin{array}{r|l}
\{-1\} & \text{features}(F(\text{pl}))(n) \\
\{-2\} & s(c) \\
\{-3\} & c(n) \\
\{-4\} & \text{satExpConsts}(F(\text{pl}),\ c) \\
\{-5\} & \text{satisfies}(\text{NOT\_FORMULA}(\text{NAME\_FORMULA}(Q)),\ c) \\
\{-6\} & \text{formulae}(F(\text{pl2})) = \text{filterFormulae}(F(\text{pl}),\ Q) \\
\{-7\} & \text{features}(F(\text{pl}))(P) \\
\{-8\} & \text{features}(F(\text{pl}))(Q) \\
\{-9\} & A(\text{pl}) = \text{overw}(\text{pairs},\ A(\text{pl2})) \\
\{-10\} & K(\text{pl2}) = (K(\text{pl}) \setminus \text{its}) \\
\{-11\} & \text{conditionsRemoveFeature}(F(\text{pl}),\ \text{its},\ \text{pairs},\ P,\ Q,\ K(\text{pl})) \\
\{-12\} & \textsc{forall}\ c\text{:}\ s(c) \Rightarrow \text{prod}(\text{pl},\ c) = \text{prod}(\text{pl2},\ c) \\
\hline
\{1\} & \text{features}(F(\text{pl2}))(n) \\
\{2\} & Q \neq n
\end{array}
$$

Trying repeated skolemization, instantiation, and if-lifting,
This completes the proof of `removeFeaturePartRefStrong.2.2`.
`removeFeaturePartRefStrong.3:`

$$
\begin{array}{r|l}
\{-1\} & \text{predRemoveFeature}(\text{pl},\ \text{pl2},\ s,\ \text{its},\ \text{pairs},\ P,\ Q) \\
\{-2\} & \textsc{forall}\ c\text{:}\ s(c) \Rightarrow \text{prod}(\text{pl},\ c) = \text{prod}(\text{pl2},\ c) \\
\hline
\{1\} & (s \subseteq \text{semantics}(F(\text{pl})))
\end{array}
$$

Expanding the definition of predRemoveFeature,
`removeFeaturePartRefStrong.3:`

$$
\begin{array}{r|l}
\{-1\} & (\text{syntaxRemoveFeature}(F(\text{pl}),\ F(\text{pl2}),\ A(\text{pl}),\ A(\text{pl2}),\ K(\text{pl}),\ K(\text{pl2}),\ P,\ Q,\ \text{its}, \\
 & \qquad \text{pairs}) \\
 & \quad \wedge \\
 & \quad \text{conditionsRemoveFeature}(F(\text{pl}),\ \text{its},\ \text{pairs},\ P,\ Q,\ K(\text{pl})) \wedge \\
 & \quad s = (\Diamond)(F(\text{pl}),\ \text{NOT\_FORMULA}(\text{NAME\_FORMULA}(Q)))) \\
\{-2\} & \textsc{forall}\ c\text{:}\ s(c) \Rightarrow \text{prod}(\text{pl},\ c) = \text{prod}(\text{pl2},\ c) \\
\hline
\{1\} & (s \subseteq \text{semantics}(F(\text{pl})))
\end{array}
$$

Applying bddsimp,
`removeFeaturePartRefStrong.3:`

$$
\begin{array}{r|l}
\{-1\} & \text{syntaxRemoveFeature}(F(\text{pl}),\ F(\text{pl2}),\ A(\text{pl}),\ A(\text{pl2}),\ K(\text{pl}),\ K(\text{pl2}),\ P,\ Q,\ \text{its}, \\
 & \qquad \text{pairs}) \\
\{-2\} & \text{conditionsRemoveFeature}(F(\text{pl}),\ \text{its},\ \text{pairs},\ P,\ Q,\ K(\text{pl})) \\
\{-3\} & s = (\Diamond)(F(\text{pl}),\ \text{NOT\_FORMULA}(\text{NAME\_FORMULA}(Q))) \\
\{-4\} & \textsc{forall}\ c\text{:}\ s(c) \Rightarrow \text{prod}(\text{pl},\ c) = \text{prod}(\text{pl2},\ c) \\
\hline
\{1\} & (s \subseteq \text{semantics}(F(\text{pl})))
\end{array}
$$

Expanding the definition of ¡¿,

`removeFeaturePartRefStrong.3:`

{-1} syntaxRemoveFeature(F(pl), F(pl2), A(pl), A(pl2), K(pl), K(pl2), P, Q, its, pairs)
{-2} conditionsRemoveFeature(F(pl), its, pairs, P, Q, K(pl))
{-3} s = ({c | semantics(F(pl))(c) $\wedge$ satisfies(NOT_FORMULA(NAME_FORMULA(Q)),
{-4} FORALL c: $s(c)$ $\Rightarrow$ prod(pl, c) = prod(pl2, c)

{1} ($s \subseteq$ semantics(F(pl)))

Applying decompose-equality,
`removeFeaturePartRefStrong.3:`

{-1} $\forall$ (x: Configuration):
$s(x)$ = (semantics(F(pl))(x) $\wedge$ satisfies(NOT_FORMULA(NAME_FORMULA(Q)),
{-2} syntaxRemoveFeature(F(pl), F(pl2), A(pl), A(pl2), K(pl), K(pl2), P, Q, its, pairs)
{-3} conditionsRemoveFeature(F(pl), its, pairs, P, Q, K(pl))
{-4} FORALL c: $s(c)$ $\Rightarrow$ prod(pl, c) = prod(pl2, c)

{1} ($s \subseteq$ semantics(F(pl)))

Expanding the definition of subset?,
`removeFeaturePartRefStrong.3:`

{-1} $\forall$ (x: Configuration):
$s(x)$ = (semantics(F(pl))(x) $\wedge$ satisfies(NOT_FORMULA(NAME_FORMULA(Q)),
{-2} syntaxRemoveFeature(F(pl), F(pl2), A(pl), A(pl2), K(pl), K(pl2), P, Q, its, pairs)
{-3} conditionsRemoveFeature(F(pl), its, pairs, P, Q, K(pl))
{-4} FORALL c: $s(c)$ $\Rightarrow$ prod(pl, c) = prod(pl2, c)

{1} FORALL (x: Configuration): ($x \in s$) $\Rightarrow$ ($x \in$ semantics(F(pl)))

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

{-1} $\forall$ (x: Configuration):
$s(x)$ = (semantics(F(pl))(x) $\wedge$ satisfies(NOT_FORMULA(NAME_FORMULA(Q)),
{-2} syntaxRemoveFeature(F(pl), F(pl2), A(pl), A(pl2), K(pl), K(pl2), P, Q, its, pairs)
{-3} conditionsRemoveFeature(F(pl), its, pairs, P, Q, K(pl))
{-4} FORALL c: $s(c)$ $\Rightarrow$ prod(pl, c) = prod(pl2, c)

{1} ($c \in s$) $\Rightarrow$ ($c \in$ semantics(F(pl)))

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

`removeFeaturePartRefStrong.3:`

{-1} $s(c) \ = \ (\text{semantics}(F(\text{pl}))(c) \ \wedge \ \text{satisfies}(\text{NOT\_FORMULA}(\text{NAME\_FORMULA}(Q)), \ c$
{-2} $\text{syntaxRemoveFeature}(F(\text{pl}), \ F(\text{pl2}), \ A(\text{pl}), \ A(\text{pl2}), \ K(\text{pl}), \ K(\text{pl2}), \ P, \ Q, \ \text{its}, \ \text{pairs})$
{-3} $\text{conditionsRemoveFeature}(F(\text{pl}), \ \text{its}, \ \text{pairs}, \ P, \ Q, \ K(\text{pl}))$
{-4} FORALL $c\text{:} \ s(c) \ \Rightarrow \ \text{prod}(\text{pl}, \ c) \ = \ \text{prod}(\text{pl2}, \ c)$

{1} $(c \in s) \ \Rightarrow \ (c \in \text{semantics}(F(\text{pl})))$

Expanding the definition of member,
`removeFeaturePartRefStrong.3:`

{-1} $s(c) \ = \ (\text{semantics}(F(\text{pl}))(c) \ \wedge \ \text{satisfies}(\text{NOT\_FORMULA}(\text{NAME\_FORMULA}(Q)), \ c$
{-2} $\text{syntaxRemoveFeature}(F(\text{pl}), \ F(\text{pl2}), \ A(\text{pl}), \ A(\text{pl2}), \ K(\text{pl}), \ K(\text{pl2}), \ P, \ Q, \ \text{its}, \ \text{pairs})$
{-3} $\text{conditionsRemoveFeature}(F(\text{pl}), \ \text{its}, \ \text{pairs}, \ P, \ Q, \ K(\text{pl}))$
{-4} FORALL $c\text{:} \ s(c) \ \Rightarrow \ \text{prod}(\text{pl}, \ c) \ = \ \text{prod}(\text{pl2}, \ c)$

{1} $s(c) \ \Rightarrow \ \text{semantics}(F(\text{pl}))(c)$

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