Contents

1 mmFoundation Theory

Built: 19 August 2017
Parent Theories: fixedPoint, indexedLists, patternMatches

1.1 Datatypes

```
mmForm =
    tt
  | ff
  | propmm 'propvar
  | (andmm) (('action, 'propvar) mmForm)
            (('action, 'propvar) mmForm)
  | (ormm) (('action, 'propvar) mmForm)
           (('action, 'propvar) mmForm)
  | Box ('action -> bool) (('action, 'propvar) mmForm)
  | Dia ('action -> bool) (('action, 'propvar) mmForm)
  | nu 'propvar (('action, 'propvar) mmForm)
  | mu 'propvar (('action, 'propvar) mmForm)
```

1.2 Definitions

[extends_def]

$\vdash \forall V\ V'.\ \texttt{extends}\ V\ V' \iff \forall Z.\ V\ Z \subseteq V'\ Z$

[mmfn_def]

$\vdash \forall \mathit{Trans}\ f\ E\ V.$
$\quad \texttt{mmfn}\ \mathit{Trans}\ f\ E\ V = \{s \mid s \in E \land (s,\mathit{Trans},V)\ \texttt{mmsat}\ f\}$

[mmUpdate_def]

$\vdash \forall Z\ V\ E\ Y.\ \texttt{mmUpdate}\ Z\ V\ E\ Y = \textbf{if}\ Y = Z\ \textbf{then}\ E\ \textbf{else}\ V\ Y$

[satFun_def]

$\vdash \forall \mathit{Trans}\ Z\ V\ \mathit{form}\ E.$
$\quad \texttt{satFun}\ \mathit{Trans}\ Z\ V\ \mathit{form}\ E =$
$\quad \texttt{mmfn}\ \mathit{Trans}\ \mathit{form}\ \mathcal{U}(\texttt{:'configuration})\ (\texttt{mmUpdate}\ Z\ V\ E)$

1.3 Theorems

[IN_CLAUSES]

$\vdash (\{s \mid s \in (\lambda x.\ P\ x \lor Q\ x)\} =$
$\quad \{s \mid s \in (\lambda x.\ P\ x) \lor s \in (\lambda x.\ Q\ x)\}) \land$
$\quad (\{s \mid s \in (\lambda x.\ P\ x \land Q\ x)\} =$
$\quad \{s \mid s \in (\lambda x.\ P\ x) \land s \in (\lambda x.\ Q\ x)\})$

[IN_UNION_INTER_CLAUSES]

$\vdash$ $(\{s \mid s \in (\lambda x.\ P\ x \land Q\ x)\} = (\lambda x.\ P\ x) \cap (\lambda x.\ Q\ x)) \land$
$(\{s \mid s \in (\lambda x.\ P\ x \lor Q\ x)\} = (\lambda x.\ P\ x) \cup (\lambda x.\ Q\ x))$

[mmfn_CLAUSES]

$\vdash$ $(\forall f_1\ f_2\ V\ \mathit{Trans}.$
$\quad \texttt{mmfn}\ \mathit{Trans}\ (f_1\ \texttt{andmm}\ f_2)\ \mathcal{U}(\texttt{:'configuration})\ V =$
$\quad \texttt{mmfn}\ \mathit{Trans}\ f_1\ \mathcal{U}(\texttt{:'configuration})\ V \cap$
$\quad \texttt{mmfn}\ \mathit{Trans}\ f_2\ \mathcal{U}(\texttt{:'configuration})\ V) \land$
$\forall f_1\ f_2\ V\ \mathit{Trans}.$
$\quad \texttt{mmfn}\ \mathit{Trans}\ (f_1\ \texttt{ormm}\ f_2)\ \mathcal{U}(\texttt{:'configuration})\ V =$
$\quad \texttt{mmfn}\ \mathit{Trans}\ f_1\ \mathcal{U}(\texttt{:'configuration})\ V \cup$
$\quad \texttt{mmfn}\ \mathit{Trans}\ f_2\ \mathcal{U}(\texttt{:'configuration})\ V$

[mmfn_MONOTONIC]

$\vdash$ $\forall \mathit{form}\ V\ V'.$
$\quad \texttt{extends}\ V\ V' \Rightarrow$
$\quad \texttt{mmfn}\ \mathit{Trans}\ \mathit{form}\ \mathcal{U}(\texttt{:'configuration})\ V \subseteq$
$\quad \texttt{mmfn}\ \mathit{Trans}\ \mathit{form}\ \mathcal{U}(\texttt{:'configuration})\ V'$

[mmfn_MONOTONIC_andmm]

$\vdash$ $(\forall V\ V'.$
$\quad \texttt{extends}\ V\ V' \Rightarrow$
$\quad \texttt{mmfn}\ \mathit{Trans}\ \mathit{form}\ \mathcal{U}(\texttt{:'configuration})\ V \subseteq$
$\quad \texttt{mmfn}\ \mathit{Trans}\ \mathit{form}\ \mathcal{U}(\texttt{:'configuration})\ V') \Rightarrow$
$(\forall V\ V'.$
$\quad \texttt{extends}\ V\ V' \Rightarrow$
$\quad \texttt{mmfn}\ \mathit{Trans}\ \mathit{form}'\ \mathcal{U}(\texttt{:'configuration})\ V \subseteq$
$\quad \texttt{mmfn}\ \mathit{Trans}\ \mathit{form}'\ \mathcal{U}(\texttt{:'configuration})\ V') \Rightarrow$
$\texttt{extends}\ V\ V' \Rightarrow$
$\texttt{mmfn}\ \mathit{Trans}\ \mathit{form}\ \mathcal{U}(\texttt{:'configuration})\ V \cap$
$\texttt{mmfn}\ \mathit{Trans}\ \mathit{form}'\ \mathcal{U}(\texttt{:'configuration})\ V \subseteq$
$\texttt{mmfn}\ \mathit{Trans}\ \mathit{form}\ \mathcal{U}(\texttt{:'configuration})\ V' \cap$
$\texttt{mmfn}\ \mathit{Trans}\ \mathit{form}'\ \mathcal{U}(\texttt{:'configuration})\ V'$

[mmfn_MONOTONIC_Box]

$\vdash$ $(\forall V\ V'.$
$\quad \texttt{extends}\ V\ V' \Rightarrow$
$\quad \texttt{mmfn}\ \mathit{Trans}\ \mathit{form}\ \mathcal{U}(\texttt{:'configuration})\ V \subseteq$
$\quad \texttt{mmfn}\ \mathit{Trans}\ \mathit{form}\ \mathcal{U}(\texttt{:'configuration})\ V') \Rightarrow$
$\texttt{extends}\ V\ V' \Rightarrow$
$\texttt{mmfn}\ \mathit{Trans}\ (\texttt{Box}\ f\ \mathit{form})\ \mathcal{U}(\texttt{:'configuration})\ V \subseteq$
$\texttt{mmfn}\ \mathit{Trans}\ (\texttt{Box}\ f\ \mathit{form})\ \mathcal{U}(\texttt{:'configuration})\ V'$

[mmfn_MONOTONIC_Dia]

```
⊢ (∀ V  V′.
      extends V  V′ ⇒
      mmfn Trans form 𝒰(:'configuration) V ⊆
      mmfn Trans form 𝒰(:'configuration) V′) ⇒
   extends V  V′ ⇒
   mmfn Trans (Dia f form) 𝒰(:'configuration) V ⊆
   mmfn Trans (Dia f form) 𝒰(:'configuration) V′
```

[mmfn_MONOTONIC_mu]

```
⊢ (∀ V  V′.
      extends V  V′ ⇒
      mmfn Trans form 𝒰(:'configuration) V ⊆
      mmfn Trans form 𝒰(:'configuration) V′) ⇒
   extends V  V′ ⇒
   mmfn Trans (mu p form) 𝒰(:'configuration) V ⊆
   mmfn Trans (mu p form) 𝒰(:'configuration) V′
```

[mmfn_MONOTONIC_nu]

```
⊢ (∀ V  V′.
      extends V  V′ ⇒
      mmfn Trans form 𝒰(:'configuration) V ⊆
      mmfn Trans form 𝒰(:'configuration) V′) ⇒
   extends V  V′ ⇒
   mmfn Trans (nu p form) 𝒰(:'configuration) V ⊆
   mmfn Trans (nu p form) 𝒰(:'configuration) V′
```

[mmfn_MONOTONIC_ormm]

```
⊢ (∀ V  V′.
      extends V  V′ ⇒
      mmfn Trans form 𝒰(:'configuration) V ⊆
      mmfn Trans form 𝒰(:'configuration) V′) ⇒
   (∀ V  V′.
      extends V  V′ ⇒
      mmfn Trans form′ 𝒰(:'configuration) V ⊆
      mmfn Trans form′ 𝒰(:'configuration) V′) ⇒
   extends V  V′ ⇒
   mmfn Trans form 𝒰(:'configuration) V ∪
   mmfn Trans form′ 𝒰(:'configuration) V ⊆
   mmfn Trans form 𝒰(:'configuration) V′ ∪
   mmfn Trans form′ 𝒰(:'configuration) V′
```

[mmfn_MONOTONIC_propvar]

```
⊢ ∀ Z  V  V′.
     extends V  V′ ⇒
     mmfn Trans (propmm Z) 𝒰(:'configuration) V ⊆
     mmfn Trans (propmm Z) 𝒰(:'configuration) V′
```

[mmfn_tt_ff_CLAUSES]

$$\begin{array}{l}
\vdash (\forall \mathit{Trans}\ V\ V'.\\
\quad\quad \texttt{mmfn}\ \mathit{Trans}\ \texttt{tt}\ \mathcal{U}(\texttt{:'configuration})\ V \subseteq\\
\quad\quad \texttt{mmfn}\ \mathit{Trans}\ \texttt{tt}\ \mathcal{U}(\texttt{:'configuration})\ V') \land\\
\quad \forall \mathit{Trans}\ V\ V'.\\
\quad\quad \texttt{mmfn}\ \mathit{Trans}\ \texttt{ff}\ \mathcal{U}(\texttt{:'configuration})\ V \subseteq\\
\quad\quad \texttt{mmfn}\ \mathit{Trans}\ \texttt{ff}\ \mathcal{U}(\texttt{:'configuration})\ V'
\end{array}$$

[mmsat_def]

$$\begin{array}{l}
\vdash (\forall V\ \mathit{Trans}\ E.\ (E,\mathit{Trans},V)\ \texttt{mmsat}\ \texttt{tt} \iff \texttt{T}) \land\\
\quad (\forall V\ \mathit{Trans}\ E.\ (E,\mathit{Trans},V)\ \texttt{mmsat}\ \texttt{ff} \iff \texttt{F}) \land\\
\quad (\forall Z\ V\ \mathit{Trans}\ E.\ (E,\mathit{Trans},V)\ \texttt{mmsat}\ \texttt{propmm}\ Z \iff E \in V\ Z) \land\\
\quad (\forall f_2\ f_1\ V\ \mathit{Trans}\ E.\\
\quad\quad (E,\mathit{Trans},V)\ \texttt{mmsat}\ f_1\ \texttt{andmm}\ f_2 \iff\\
\quad\quad (E,\mathit{Trans},V)\ \texttt{mmsat}\ f_1 \land (E,\mathit{Trans},V)\ \texttt{mmsat}\ f_2) \land\\
\quad (\forall f_2\ f_1\ V\ \mathit{Trans}\ E.\\
\quad\quad (E,\mathit{Trans},V)\ \texttt{mmsat}\ f_1\ \texttt{ormm}\ f_2 \iff\\
\quad\quad (E,\mathit{Trans},V)\ \texttt{mmsat}\ f_1 \lor (E,\mathit{Trans},V)\ \texttt{mmsat}\ f_2) \land\\
\quad (\forall f\ V\ \mathit{Trans}\ E\ \mathit{Actions}.\\
\quad\quad (E,\mathit{Trans},V)\ \texttt{mmsat}\ \texttt{Box}\ \mathit{Actions}\ f \iff\\
\quad\quad \forall E'\ a.\\
\quad\quad\quad \mathit{Trans}\ a\ E\ E' \Rightarrow a \in \mathit{Actions} \Rightarrow (E',\mathit{Trans},V)\ \texttt{mmsat}\ f) \land\\
\quad (\forall f\ V\ \mathit{Trans}\ E\ \mathit{Actions}.\\
\quad\quad (E,\mathit{Trans},V)\ \texttt{mmsat}\ \texttt{Dia}\ \mathit{Actions}\ f \iff\\
\quad\quad \exists E'\ a.\\
\quad\quad\quad \mathit{Trans}\ a\ E\ E' \land a \in \mathit{Actions} \land (E',\mathit{Trans},V)\ \texttt{mmsat}\ f) \land\\
\quad (\forall f\ Z\ V\ \mathit{Trans}\ E.\\
\quad\quad (E,\mathit{Trans},V)\ \texttt{mmsat}\ \texttt{nu}\ Z\ f \iff\\
\quad\quad \exists \mathit{setE}.\\
\quad\quad\quad E \in \mathit{setE} \land\\
\quad\quad\quad \forall E'.\ E' \in \mathit{setE} \Rightarrow (E',\mathit{Trans},\texttt{mmUpdate}\ Z\ V\ \mathit{setE})\ \texttt{mmsat}\ f) \land\\
\quad \forall f\ Z\ V\ \mathit{Trans}\ E.\\
\quad\quad (E,\mathit{Trans},V)\ \texttt{mmsat}\ \texttt{mu}\ Z\ f \iff\\
\quad\quad \forall \mathit{setE}.\\
\quad\quad\quad E \notin \mathit{setE} \Rightarrow\\
\quad\quad\quad \exists E'.\ (E',\mathit{Trans},\texttt{mmUpdate}\ Z\ V\ \mathit{setE})\ \texttt{mmsat}\ f \land E' \notin \mathit{setE}
\end{array}$$

[mmsat_IN_CLAUSES]

$$\begin{array}{l}
\vdash (\forall s\ \mathit{form}\ V\ \mathit{Trans}.\\
\quad\quad \{s \mid (s,\mathit{Trans},V)\ \texttt{mmsat}\ \mathit{form}\} =\\
\quad\quad \{s \mid s \in (\lambda x.\ (x,\mathit{Trans},V)\ \texttt{mmsat}\ \mathit{form})\}) \land\\
\quad (\forall s\ f_1\ f_2\ V.\\
\quad\quad \{s \mid (s,\mathit{Trans},V)\ \texttt{mmsat}\ f_1 \land (s,\mathit{Trans},V)\ \texttt{mmsat}\ f_2\} =\\
\quad\quad \{s \mid\\
\quad\quad\ s \in (\lambda x.\ (x,\mathit{Trans},V)\ \texttt{mmsat}\ f_1) \land\\
\quad\quad\ s \in (\lambda x.\ (x,\mathit{Trans},V)\ \texttt{mmsat}\ f_2)\}) \land\\
\quad \forall s\ f_1\ f_2\ V.\\
\quad\quad \{s \mid (s,\mathit{Trans},V)\ \texttt{mmsat}\ f_1 \lor (s,\mathit{Trans},V)\ \texttt{mmsat}\ f_2\} =
\end{array}$$

$$\{s \mid s \in (\lambda x.\ (x, \mathit{Trans}, V)\ \texttt{mmsat}\ f_1) \lor s \in (\lambda x.\ (x, \mathit{Trans}, V)\ \texttt{mmsat}\ f_2)\}$$

[mmsat_ind]

$$\begin{aligned}
\vdash \forall P.\ & \\
& (\forall E\ \mathit{Trans}\ V.\ P\ (E, \mathit{Trans}, V)\ \texttt{tt}) \land \\
& (\forall E\ \mathit{Trans}\ V.\ P\ (E, \mathit{Trans}, V)\ \texttt{ff}) \land \\
& (\forall E\ \mathit{Trans}\ V\ Z.\ P\ (E, \mathit{Trans}, V)\ (\texttt{propmm}\ Z)) \land \\
& (\forall E\ \mathit{Trans}\ V\ f_1\ f_2. \\
& \quad P\ (E, \mathit{Trans}, V)\ f_1 \land P\ (E, \mathit{Trans}, V)\ f_2 \Rightarrow \\
& \quad P\ (E, \mathit{Trans}, V)\ (f_1\ \texttt{andmm}\ f_2)) \land \\
& (\forall E\ \mathit{Trans}\ V\ f_1\ f_2. \\
& \quad P\ (E, \mathit{Trans}, V)\ f_1 \land P\ (E, \mathit{Trans}, V)\ f_2 \Rightarrow \\
& \quad P\ (E, \mathit{Trans}, V)\ (f_1\ \texttt{ormm}\ f_2)) \land \\
& (\forall E\ \mathit{Trans}\ V\ \mathit{Actions}\ f. \\
& \quad (\forall a\ E'.\ \mathit{Trans}\ a\ E\ E' \land a \in \mathit{Actions} \Rightarrow P\ (E', \mathit{Trans}, V)\ f) \Rightarrow \\
& \quad P\ (E, \mathit{Trans}, V)\ (\texttt{Box}\ \mathit{Actions}\ f)) \land \\
& (\forall E\ \mathit{Trans}\ V\ \mathit{Actions}\ f. \\
& \quad (\forall E'.\ P\ (E', \mathit{Trans}, V)\ f) \Rightarrow \\
& \quad P\ (E, \mathit{Trans}, V)\ (\texttt{Dia}\ \mathit{Actions}\ f)) \land \\
& (\forall E\ \mathit{Trans}\ V\ Z\ f. \\
& \quad (\forall E'\ \mathit{setE}. \\
& \qquad E' \in \mathit{setE} \Rightarrow P\ (E', \mathit{Trans}, \texttt{mmUpdate}\ Z\ V\ \mathit{setE})\ f) \Rightarrow \\
& \quad P\ (E, \mathit{Trans}, V)\ (\texttt{nu}\ Z\ f)) \land \\
& (\forall E\ \mathit{Trans}\ V\ Z\ f. \\
& \quad (\forall \mathit{setE}\ E'. \\
& \qquad E \notin \mathit{setE} \Rightarrow P\ (E', \mathit{Trans}, \texttt{mmUpdate}\ Z\ V\ \mathit{setE})\ f) \Rightarrow \\
& \quad P\ (E, \mathit{Trans}, V)\ (\texttt{mu}\ Z\ f)) \Rightarrow \\
& \forall v\ v_1\ v_2\ v_3.\ P\ (v, v_1, v_2)\ v_3
\end{aligned}$$

[mmsat_mu_lfp]

$$\begin{aligned}
\vdash \forall f\ Z\ V\ \mathit{Trans}\ E.\ & \\
& (E, \mathit{Trans}, V)\ \texttt{mmsat}\ \texttt{mu}\ Z\ f \iff E \in \texttt{lfp}\ (\texttt{satFun}\ \mathit{Trans}\ Z\ V\ f)
\end{aligned}$$

[mmsat_nu_gfp]

$$\begin{aligned}
\vdash \forall f\ Z\ V\ \mathit{Trans}\ E.\ & \\
& (E, \mathit{Trans}, V)\ \texttt{mmsat}\ \texttt{nu}\ Z\ f \iff E \in \texttt{gfp}\ (\texttt{satFun}\ \mathit{Trans}\ Z\ V\ f)
\end{aligned}$$

[mmUpdate_MONOTONIC]

$$\begin{aligned}
\vdash\ & (\forall V\ Z\ E\ F. \\
& \quad E \subseteq F \Rightarrow \texttt{extends}\ (\texttt{mmUpdate}\ Z\ V\ E)\ (\texttt{mmUpdate}\ Z\ V\ F)) \land \\
& \forall V\ V'\ Z\ E. \\
& \quad \texttt{extends}\ V\ V' \Rightarrow \texttt{extends}\ (\texttt{mmUpdate}\ Z\ V\ E)\ (\texttt{mmUpdate}\ Z\ V'\ E)
\end{aligned}$$

[MONOTONE_INTER]

$$\vdash A \subseteq A' \Rightarrow B \subseteq B' \Rightarrow A \cap B \subseteq A' \cap B'$$

[MONOTONE_UNION]
$\vdash A \subseteq A' \Rightarrow B \subseteq B' \Rightarrow A \cup B \subseteq A' \cup B'$

[satFun_gfp_thm]
$\vdash$ gfp (satFun *Trans* *Z* *V* *f*) =
BIGUNION $\{$ *setE* | *setE* $\subseteq$ satFun *Trans* *Z* *V* *f* *setE* $\}$

[satFun_lfp_thm]
$\vdash$ lfp (satFun *Trans* *Z* *V* *f*) =
BIGINTER $\{$ *setE* | satFun *Trans* *Z* *V* *f* *setE* $\subseteq$ *setE* $\}$

[satFun_MONOTONIC]
$\vdash \forall V$ *Trans* *Z* *form* E_1 E_2.
$E_1 \subseteq E_2 \Rightarrow$
satFun *Trans* *Z* *V* *form* $E_1 \subseteq$ satFun *Trans* *Z* *V* *form* E_2

2 substitution Theory

Built: 19 August 2017

Parent Theories: mmFoundation, res_quan

2.1 Definitions

[extend_env_def]
$\vdash \forall x\ v\ f.$ extend_env x v f = ($\lambda y.$ **if** $y = x$ **then** v **else** f y)

[fmla_size_def]
$\vdash$ (fmla_size tt = 0) $\wedge$ (fmla_size ff = 0) $\wedge$
($\forall Z.$ fmla_size (propmm Z) = 1) $\wedge$
($\forall f_1$ f_2.
fmla_size (f_1 andmm f_2) =
1 + fmla_size f_1 + fmla_size f_2) $\wedge$
($\forall f_1$ f_2.
fmla_size (f_1 ormm f_2) =
1 + fmla_size f_1 + fmla_size f_2) $\wedge$
($\forall$ *Actions* f. fmla_size (Box *Actions* f) = 1 + fmla_size f) $\wedge$
($\forall$ *Actions* f. fmla_size (Dia *Actions* f) = 1 + fmla_size f) $\wedge$
($\forall Z$ f. fmla_size (nu Z f) = 1 + fmla_size f) $\wedge$
$\forall Z$ f. fmla_size (mu Z f) = 1 + fmla_size f

[frees_def]
$\vdash$ (frees tt = $\{\}$) $\wedge$ (frees ff = $\{\}$) $\wedge$
($\forall Z.$ frees (propmm Z) = $\{Z\}$) $\wedge$
($\forall f_1$ f_2. frees (f_1 andmm f_2) = frees f_1 $\cup$ frees f_2) $\wedge$
($\forall f_1$ f_2. frees (f_1 ormm f_2) = frees f_1 $\cup$ frees f_2) $\wedge$
($\forall$ *Actions* f. frees (Box *Actions* f) = frees f) $\wedge$
($\forall$ *Actions* f. frees (Dia *Actions* f) = frees f) $\wedge$
($\forall Z$ f. frees (nu Z f) = frees f DELETE Z) $\wedge$
$\forall Z$ f. frees (mu Z f) = frees f DELETE Z

[gv_def]

$\vdash \forall l\ Z\ fs.$ `gv` $l\ Z\ fs$ = `fv (gl` $l\ Z\ fs$`)` Z

[l_Sub_def]

```
⊢ (∀l. l_Sub l tt = tt) ∧ (∀l. l_Sub l ff = ff) ∧
  (∀l Y. l_Sub l (propmm Y) = propmm (fv l Y)) ∧
  (∀l P Q. l_Sub l (P andmm Q) = l_Sub l P andmm l_Sub l Q) ∧
  (∀l P Q. l_Sub l (P ormm Q) = l_Sub l P ormm l_Sub l Q) ∧
  (∀l Actions P.
     l_Sub l (Box Actions P) = Box Actions (l_Sub l P)) ∧
  (∀l Actions P.
     l_Sub l (Dia Actions P) = Dia Actions (l_Sub l P)) ∧
  (∀l Z P.
     l_Sub l (nu Z P) =
     (let
        fs = frees P ;
        (Z',l') = (gv l Z fs,gl l Z fs)
      in
        nu Z' (l_Sub l' P))) ∧
  ∀l Z P.
    l_Sub l (mu Z P) =
    (let
       fs = frees P ;
       (Z',l') = (gv l Z fs,gl l Z fs)
     in
       mu Z' (l_Sub l' P))
```

[rf_def]

```
⊢ ∀Y X fs.
    rf Y X fs = if X ∈ fs then Y INSERT fs DELETE X else fs
```

[setsat_def]

```
⊢ ∀Trans f V. setsat Trans f V = {E | (E,Trans,V) mmsat f}
```

[variant_spec]

```
⊢ ∀exclvars.
    INFINITE 𝓤(:'variable) ⇒
    FINITE exclvars ⇒
    ∀v. variant exclvars v ∉ exclvars
```

[vars_def]

```
⊢ (vars tt = {}) ∧ (vars ff = {}) ∧
  (∀Z. vars (propmm Z) = {Z}) ∧
  (∀f1 f2. vars (f1 andmm f2) = vars f1 ∪ vars f2) ∧
  (∀f1 f2. vars (f1 ormm f2) = vars f1 ∪ vars f2) ∧
  (∀Actions f. vars (Box Actions f) = vars f) ∧
  (∀Actions f. vars (Dia Actions f) = vars f) ∧
  (∀Z f. vars (nu Z f) = vars f ∪ {Z}) ∧
  ∀Z f. vars (mu Z f) = vars f ∪ {Z}
```

2.2 Theorems

[alpha_frees]

$\vdash \forall Y\ X\ Fm.$
$\quad$ INFINITE $\mathcal{U}$(:'b) $\Rightarrow$
$\quad Y \notin$ frees $Fm \Rightarrow$
$\quad$ (frees (Subst (propmm Y) X Fm) = rf Y X (frees Fm))

[alpha_LEM]

$\vdash \forall Trans\ Fm\ V\ Q\ X\ X'.$
$\quad$ INFINITE $\mathcal{U}$(:'b) $\Rightarrow$
$\quad X' \notin$ frees $Fm \Rightarrow$
$\quad$ (setsat $Trans$ (Subst (propmm X') X Fm)
$\quad\quad$ (extend_env X' Q V) =
$\quad$ setsat $Trans$ Fm (extend_env X Q V))

[alpha_remove]

$\vdash \forall Y\ X\ Fm.$
$\quad$ INFINITE $\mathcal{U}$(:'b) $\Rightarrow$
$\quad Y \notin$ frees $Fm \land Y \neq X \Rightarrow$
$\quad X \notin$ frees (Subst (propmm Y) X Fm)

[COND_NOT]

$\vdash \forall P\ A\ B.$ (**if** $\neg P$ **then** A **else** B) = **if** P **then** B **else** A

[COND_NOT_DISJ]

$\vdash \forall P\ Q\ A\ B.$
$\quad$ (**if** $\neg Q \lor P$ **then** A **else** B) =
$\quad$ **if** P **then** A **else** **if** Q **then** B **else** A

[EQ_SUBSET_SUBSET]

$\vdash \forall s_1\ s_2.\ (s_1 = s_2) \iff s_1 \subseteq s_2 \land s_2 \subseteq s_1$

[extend_env_mmUpdate_EQ]

$\vdash$ extend_env Z E V = mmUpdate Z V E

[extend_env_mmUpdate_lemma]

$\vdash$ extend_env Z E V Y = mmUpdate Z V E Y

[fmla_size_ind]

$\vdash \forall P.$
$\quad$ ($\forall f$. ($\forall g$. fmla_size g < fmla_size $f \Rightarrow P\ g) \Rightarrow P\ f) \Rightarrow$
$\quad \forall n\ f.$ (fmla_size f = n) $\Rightarrow P\ f$

[fmla_size_induction]

$\vdash \forall P.$
$\quad P\ \mathtt{tt} \land P\ \mathtt{ff} \land (\forall s.\ P\ (\mathtt{propmm}\ s)) \land$
$\quad (\forall f\ g.\ P\ f \land P\ g \Rightarrow P\ (f\ \mathtt{andmm}\ g)) \land$
$\quad (\forall f\ g.\ P\ f \land P\ g \Rightarrow P\ (f\ \mathtt{ormm}\ g)) \land$
$\quad (\forall \mathit{Actions}\ f.\ P\ f \Rightarrow P\ (\mathtt{Box}\ \mathit{Actions}\ f)) \land$
$\quad (\forall \mathit{Actions}\ f.\ P\ f \Rightarrow P\ (\mathtt{Dia}\ \mathit{Actions}\ f)) \land$
$\quad (\forall f.$
$\qquad (\forall g.\ (\mathtt{fmla\_size}\ g = \mathtt{fmla\_size}\ f) \Rightarrow P\ g) \Rightarrow$
$\qquad \forall s.\ P\ (\mathtt{nu}\ s\ f)) \land$
$\quad (\forall f.$
$\qquad (\forall g.\ (\mathtt{fmla\_size}\ g = \mathtt{fmla\_size}\ f) \Rightarrow P\ g) \Rightarrow$
$\qquad \forall s.\ P\ (\mathtt{mu}\ s\ f)) \Rightarrow$
$\quad \forall f.\ P\ f$

[frees_are_vars]

$\vdash \forall f\ x.\ x \in \mathtt{frees}\ f \Rightarrow x \in \mathtt{vars}\ f$

[frees_finite]

$\vdash \forall f.\ \mathtt{FINITE}\ (\mathtt{frees}\ f)$

[frees_LEM]

$\vdash \forall \mathit{Fm}\ l.$
$\quad \mathtt{INFINITE}\ \mathcal{U}(\mathtt{:'b}) \Rightarrow$
$\quad \mathtt{ok\_r}\ l\ (\mathtt{frees}\ \mathit{Fm}) \Rightarrow$
$\quad (\mathtt{frees}\ (\mathtt{l\_Sub}\ l\ \mathit{Fm}) = \mathtt{gf}\ l\ (\mathtt{frees}\ \mathit{Fm}))$

[frees_SUBSET_vars]

$\vdash \forall f.\ \mathtt{frees}\ f \subseteq \mathtt{vars}\ f$

[fv_1_1]

$\vdash \forall l\ \mathit{fs}.$
$\quad \mathtt{ok\_r}\ l\ \mathit{fs} \Rightarrow$
$\quad \forall A\ B.\ A \in \mathit{fs} \land B \in \mathit{fs} \Rightarrow ((\mathtt{fv}\ l\ A = \mathtt{fv}\ l\ B) \iff (A = B))$

[fv_append]

$\vdash \forall l\ m.\ \mathtt{fv}\ (l\ \mathtt{++}\ m) = \mathtt{fv}\ l \circ \mathtt{fv}\ m$

[fv_BIJ]

$\vdash \forall l\ \mathit{fs}.\ \mathtt{ok\_r}\ l\ \mathit{fs} \Rightarrow \mathtt{BIJ}\ (\mathtt{fv}\ l)\ \mathit{fs}\ (\mathtt{gf}\ l\ \mathit{fs})$

[fv_def]

$\vdash (\forall X.\ \mathtt{fv}\ []\ X = X) \land$
$\quad \forall l\ Z\ Y\ X.$
$\qquad \mathtt{fv}\ ((Y,Z)\mathtt{::}l)\ X =$
$\qquad (\mathbf{let}\ X' = \mathtt{fv}\ l\ X\ \mathbf{in}\ \mathbf{if}\ X' = Y\ \mathbf{then}\ Z\ \mathbf{else}\ X')$

[fv_IN_gf]

$\vdash \forall l\ fs\ (A\texttt{::}fs).\ \texttt{fv}\ l\ A \in \texttt{gf}\ l\ fs$

[fv_ind]

$\vdash \forall P.$
$\quad (\forall X.\ P\ \texttt{[]}\ X) \land (\forall Y\ Z\ l\ X.\ P\ l\ X \Rightarrow P\ ((Y,Z)\texttt{::}l)\ X) \Rightarrow$
$\quad \forall v\ v_1.\ P\ v\ v_1$

[fv_inj]

$\vdash \forall l\ fs.$
$\quad \texttt{ok\_r}\ l\ fs \Rightarrow$
$\quad \forall A\ B.\ A \in fs \land B \in fs \Rightarrow (\texttt{fv}\ l\ A = \texttt{fv}\ l\ B) \Rightarrow (A = B)$

[fv_LEM]

$\vdash \forall l\ s.\ \texttt{gf}\ l\ \{s\} = \{\texttt{fv}\ l\ s\}$

[fv_not_in]

$\vdash \forall fs\ gs\ Z\ l.$
$\quad \texttt{ok\_r}\ l\ fs \land gs \subseteq fs \land Z \in fs \land Z \notin gs \Rightarrow \texttt{fv}\ l\ Z \notin \texttt{gf}\ l\ gs$

[gf_append]

$\vdash \forall l\ m\ fs.\ \texttt{gf}\ (l\ \texttt{++}\ m)\ fs = \texttt{gf}\ l\ (\texttt{gf}\ m\ fs)$

[gf_def]

$\vdash (\forall fs.\ \texttt{gf}\ \texttt{[]}\ fs = fs) \land$
$\quad \forall l\ fs\ Y\ X.\ \texttt{gf}\ ((X,Y)\texttt{::}l)\ fs = \texttt{rf}\ Y\ X\ (\texttt{gf}\ l\ fs)$

[gf_delete]

$\vdash \forall l\ fs\ Z.$
$\quad \texttt{ok\_r}\ l\ (Z\ \texttt{INSERT}\ fs) \Rightarrow$
$\quad (\texttt{gf}\ l\ (fs\ \texttt{DELETE}\ Z) = \texttt{gf}\ l\ fs\ \texttt{DELETE}\ \texttt{fv}\ l\ Z)$

[gf_empty]

$\vdash \forall l.\ \texttt{gf}\ l\ \{\} = \{\}$

[gf_finite]

$\vdash \forall fs.\ \texttt{FINITE}\ fs \Rightarrow \forall l.\ \texttt{FINITE}\ (\texttt{gf}\ l\ fs)$

[gf_im]

$\vdash \forall l.\ \texttt{gf}\ l = \texttt{IMAGE}\ (\texttt{fv}\ l)$

[gf_ind]

$\vdash \forall P.$
$\quad (\forall fs.\ P\ \texttt{[]}\ fs) \land (\forall X\ Y\ l\ fs.\ P\ l\ fs \Rightarrow P\ ((X,Y)\texttt{::}l)\ fs) \Rightarrow$
$\quad \forall v\ v_1.\ P\ v\ v_1$

[gf_insert]

$\vdash \forall l\ fs\ Z.\ \mathtt{gf}\ l\ (Z\ \mathtt{INSERT}\ fs) = \mathtt{fv}\ l\ Z\ \mathtt{INSERT}\ \mathtt{gf}\ l\ fs$

[gf_monotone]

$\vdash \forall l\ big\ sma.\ sma \subseteq big \Rightarrow \mathtt{gf}\ l\ sma \subseteq \mathtt{gf}\ l\ big$

[gf_union]

$\vdash \forall l\ fs\ fs'.\ \mathtt{gf}\ l\ (fs \cup fs') = \mathtt{gf}\ l\ fs \cup \mathtt{gf}\ l\ fs'$

[gfp_monotone]

$\vdash \forall G\ H.\ (\forall s.\ G\ s \subseteq H\ s) \Rightarrow \mathtt{gfp}\ G \subseteq \mathtt{gfp}\ H$

[gl_append]

$\vdash \forall Z\ fs\ m\ l.$
$\quad \mathtt{gl}\ (l\ \mathtt{++}\ m)\ Z\ fs =$
$\quad \mathtt{gl}\ l\ (\mathtt{fv}\ (\mathtt{gl}\ m\ Z\ fs)\ Z)\ (\mathtt{gf}\ (\mathtt{gl}\ m\ Z\ fs)\ fs)\ \mathtt{++}\ \mathtt{gl}\ m\ Z\ fs$

[gl_def]

$\vdash (\forall fs\ Z.\ \mathtt{gl}\ []\ Z\ fs = []) \land$
$\quad \forall l\ fs\ Z\ Y\ X.$
$\quad\quad \mathtt{gl}\ ((X,Y)::l)\ Z\ fs =$
$\quad\quad (\mathbf{let}$
$\quad\quad\quad l' = \mathtt{gl}\ l\ Z\ fs\ ;$
$\quad\quad\quad (fs',Z') = (\mathtt{gf}\ l'\ fs, \mathtt{fv}\ l'\ Z)$
$\quad\quad \mathbf{in}$
$\quad\quad\quad \mathbf{if}\ X \notin fs' \lor (X = Z')\ \mathbf{then}\ l'$
$\quad\quad\quad \mathbf{else\ if}\ Y = Z'\ \mathbf{then}$
$\quad\quad\quad\quad (X,Y)::(Z',\mathtt{variant}\ (Y\ \mathtt{INSERT}\ fs')\ Z')::l'$
$\quad\quad\quad \mathbf{else}\ (X,Y)::l')$

[gl_ind]

$\vdash \forall P.$
$\quad (\forall Z\ fs.\ P\ []\ Z\ fs) \land$
$\quad (\forall X\ Y\ l\ Z\ fs.\ P\ l\ Z\ fs \Rightarrow P\ ((X,Y)::l)\ Z\ fs) \Rightarrow$
$\quad \forall v\ v_1\ v_2.\ P\ v\ v_1\ v_2$

[half_gl_ok]

$\vdash \forall l\ Z\ fs.$
$\quad \mathtt{INFINITE}\ \mathcal{U}(:\text{'a}) \land \mathtt{FINITE}\ fs \land \mathtt{ok\_r}\ l\ (fs\ \mathtt{DELETE}\ Z) \Rightarrow$
$\quad \mathtt{ok\_r}\ (\mathtt{gl}\ l\ Z\ fs)\ (Z\ \mathtt{INSERT}\ fs)$

[in_not_in_not_eq]

$\vdash \forall X\ Y\ s.\ X \in s \land Y \notin s \Rightarrow X \neq Y$

[INSERT_INSERT_DELETE]

$\vdash \forall a\ t.\ a\ \mathtt{INSERT}\ t\ \mathtt{DELETE}\ a = a\ \mathtt{INSERT}\ t$

[l_Sub_append]

$\vdash \forall P\ l\ m.$
  INFINITE $\mathcal{U}$(:'b) $\Rightarrow$
  ok_r m (frees P) $\Rightarrow$
  (l_Sub (l ++ m) P = l_Sub l (l_Sub m P))

[l_Sub_ID]

$\vdash \forall Fm.$ l_Sub [(X,X)] Fm = Fm

[l_Sub_ID_CONS]

$\vdash \forall f\ l.$ l_Sub ((X,X)::l) f = l_Sub l f

[l_Sub_nil]

$\vdash \forall Fm.$ l_Sub [] Fm = Fm

[l_Sub_same_size]

$\vdash \forall Fm\ l.$ fmla_size (l_Sub l Fm) = fmla_size Fm

[last_extension_counts]

$\vdash \forall x\ v\ v'\ f.$
  extend_env x v (extend_env x v' f) = extend_env x v f

[last_update_counts]

$\vdash \forall x\ v\ v'\ f.$ mmUpdate x (mmUpdate x f v') v = mmUpdate x f v

[lfp_monotone]

$\vdash \forall G\ H.$ ($\forall s.$ G s $\subseteq$ H s) $\Rightarrow$ lfp G $\subseteq$ lfp H

[mmsat_setsat]

$\vdash$ (E,$Trans$,V) mmsat f $\iff$ E $\in$ setsat $Trans$ f V

[muvar_not_free]

$\vdash \forall s\ Fm.$ $s \notin$ frees (mu s Fm)

[not_in_gf]

$\vdash \forall A\ excl\ l\ fs\ Q.$
  INFINITE $\mathcal{U}$(:'a) $\Rightarrow$
  FINITE $excl$ $\Rightarrow$
  $A \notin$ gf ((A,variant (A INSERT $excl$) Q)::l) fs

[nuvar_not_free]

$\vdash \forall s\ Fm.$ $s \notin$ frees (nu s Fm)

[ok_r_def]

$\vdash$ ($\forall fs$. ok_r [] fs $\iff$ T) $\land$
$\forall l\ fs\ Y\ X$.
ok_r ((X,Y)::l) fs $\iff$
ok_r l fs $\land$ ($X \in$ gf l fs $\Rightarrow$ $Y \notin$ gf l fs)

[ok_r_gl_insert]

$\vdash \forall l\ Z\ fs$.
INFINITE $\mathcal{U}$(:'a) $\land$ FINITE fs $\land$ ok_r l (fs DELETE Z) $\Rightarrow$
ok_r (gl l Z fs) (Z INSERT fs) $\land$
$\forall X$::fs DELETE Z. fv (gl l Z fs) X = fv l X

[ok_r_ind]

$\vdash \forall P$.
($\forall fs$. P [] fs) $\land$ ($\forall X\ Y\ l\ fs$. P l fs $\Rightarrow$ P ((X,Y)::l) fs) $\Rightarrow$
$\forall v\ v_1$. P v v_1

[ok_r_subset]

$\vdash \forall l\ big\ sma$. $sma \subseteq big$ $\Rightarrow$ ok_r l big $\Rightarrow$ ok_r l sma

[ok_to_unroll_mu]

$\vdash \forall Trans\ Fm\ Z\ V$.
INFINITE $\mathcal{U}$(:'b) $\Rightarrow$
(setsat $Trans$ (Subst (mu Z Fm) Z Fm) V =
 setsat $Trans$ (mu Z Fm) V)

[ok_to_unroll_nu]

$\vdash \forall Trans\ Fm\ Z\ V$.
INFINITE $\mathcal{U}$(:'b) $\Rightarrow$
(setsat $Trans$ (Subst (nu Z Fm) Z Fm) V =
 setsat $Trans$ (nu Z Fm) V)

[pair_list_induction]

$\vdash \forall P$. P [] $\land$ ($\forall l$. P l $\Rightarrow$ $\forall X\ Y$. P ((X,Y)::l)) $\Rightarrow$ $\forall l$. P l

[setsat_EQ_satFun]

$\vdash \forall Trans\ Fm\ Z\ E\ V$.
setsat $Trans$ Fm (extend_env Z E V) = satFun $Trans$ Z V Fm E

[setsat_is_mmfn_UNIV]

$\vdash$ setsat $Trans$ f V = mmfn $Trans$ f $\mathcal{U}$(:'a) V

[setsat_lemma]

```
⊢ (∀ V. setsat Trans tt V = 𝒰(:'configuration)) ∧
  (∀ V. setsat Trans ff V = { }) ∧
  (∀ Z V. setsat Trans (propmm Z) V = V Z) ∧
  (∀ Fm1 Fm2 V.
     setsat Trans (Fm1 andmm Fm2) V =
     setsat Trans Fm1 V ∩ setsat Trans Fm2 V) ∧
  (∀ Fm1 Fm2 V.
     setsat Trans (Fm1 ormm Fm2) V =
     setsat Trans Fm1 V ∪ setsat Trans Fm2 V) ∧
  (∀ Z Fm V.
     setsat Trans (Box Actions Fm) V =
     {E |
      ∀ E' a.
        Trans a E E' ⇒ a ∈ Actions ⇒ (E',Trans,V) mmsat Fm}) ∧
  (∀ Z Fm V.
     setsat Trans (Dia Actions Fm) V =
     {E |
      ∃ E' a.
        Trans a E E' ∧ a ∈ Actions ∧ (E',Trans,V) mmsat Fm}) ∧
  (∀ Z Fm V.
     setsat Trans (nu Z Fm) V =
     gfp (λ Q. setsat Trans Fm (extend_env Z Q V))) ∧
  ∀ Z Fm V.
    setsat Trans (mu Z Fm) V =
    lfp (λ Q. setsat Trans Fm (extend_env Z Q V))
```

[setsat_monotone]

```
⊢ ∀ Trans Fm Z V.
    monotone (λ Q. setsat Trans Fm (extend_env Z Q V))
```

[silly_extend]

```
⊢ ∀ Trans Z Fm a V.
    Z ∉ frees Fm ⇒
    (setsat Trans Fm (extend_env Z a V) = setsat Trans Fm V)
```

[simple_ok_r_gl]

```
⊢ (∀ l s Fm.
     INFINITE 𝒰(:'b) ∧ ok_r l (frees (nu s Fm)) ⇒
     ok_r (gl l s (frees Fm)) (frees Fm)) ∧
  ∀ l s Fm.
    INFINITE 𝒰(:'b) ∧ ok_r l (frees (mu s Fm)) ⇒
    ok_r (gl l s (frees Fm)) (frees Fm)
```

[simple_ok_r_gl_mu]

```
⊢ ∀ l s Fm.
    INFINITE 𝒰(:'b) ∧ ok_r l (frees (mu s Fm)) ⇒
    ok_r (gl l s (frees Fm)) (frees Fm)
```

[simple_ok_r_gl_nu]

```
⊢ ∀l s Fm.
     INFINITE 𝒰(:'b) ∧ ok_r l (frees (nu s Fm)) ⇒
     ok_r (gl l s (frees Fm)) (frees Fm)
```

[Subst]

```
⊢ INFINITE 𝒰(:'b) ⇒
  (Subst p X tt = tt) ∧ (Subst p X ff = ff) ∧
  (Subst p X (propmm Z) = if Z = X then p else propmm Z) ∧
  (Subst p X (Fm₁ andmm Fm₂) =
   Subst p X Fm₁ andmm Subst p X Fm₂) ∧
  (Subst p X (Fm₁ ormm Fm₂) =
   Subst p X Fm₁ ormm Subst p X Fm₂) ∧
  (Subst p X (Box Actions Fm) = Box Actions (Subst p X Fm)) ∧
  (Subst p X (Dia Actions Fm) = Dia Actions (Subst p X Fm)) ∧
  (Subst p X (nu Z Fm) =
   (let
      fs = frees Fm
    in
      if X ∉ frees (nu Z Fm) then nu Z Fm
      else if Z ∈ frees p then
        (let
           Z' = variant (frees p ∪ fs) Z
         in
           nu Z' (Subst p X (Subst (propmm Z') Z Fm)))
      else nu Z (Subst p X Fm))) ∧
  (Subst p X (mu Z Fm) =
   (let
      fs = frees Fm
    in
      if X ∉ frees (mu Z Fm) then mu Z Fm
      else if Z ∈ frees p then
        (let
           Z' = variant (frees p ∪ fs) Z
         in
           mu Z' (Subst p X (Subst (propmm Z') Z Fm)))
      else mu Z (Subst p X Fm)))
```

[Subst_def]

```
⊢ (Subst N X tt = tt) ∧ (Subst N X ff = ff) ∧
  (Subst N X (propmm Y) = if Y = X then N else propmm Y) ∧
  (Subst N X (P andmm Q) = Subst N X P andmm Subst N X Q) ∧
  (Subst N X (P ormm Q) = Subst N X P ormm Subst N X Q) ∧
  (Subst N X (Box Actions P) = Box Actions (Subst N X P)) ∧
  (Subst N X (Dia Actions P) = Dia Actions (Subst N X P)) ∧
  (Subst N X (nu Z P) =
   (let
      fs = frees P
```

```
      in
        if X ∉ fs ∨ (X = Z) then nu Z P
        else if Z ∈ frees N then
          (let
             W = variant (frees N ∪ fs) Z
           in
             nu W (Subst N X (l_Sub [(Z,W)] P)))
        else nu Z (Subst N X P))) ∧
    (Subst N X (mu Z P) =
     (let
        fs = frees P
      in
        if X ∉ fs ∨ (X = Z) then mu Z P
        else if Z ∈ frees N then
          (let
             W = variant (frees N ∪ fs) Z
           in
             mu W (Subst N X (l_Sub [(Z,W)] P)))
        else mu Z (Subst N X P)))
```

[Subst_ind]

```
⊢ ∀P′.
    (∀N X. P′ N X tt) ∧ (∀N X. P′ N X ff) ∧
    (∀N X Y. P′ N X (propmm Y)) ∧
    (∀N X P Q. P′ N X P ∧ P′ N X Q ⇒ P′ N X (P andmm Q)) ∧
    (∀N X P Q. P′ N X P ∧ P′ N X Q ⇒ P′ N X (P ormm Q)) ∧
    (∀N X Actions P. P′ N X P ⇒ P′ N X (Box Actions P)) ∧
    (∀N X Actions P. P′ N X P ⇒ P′ N X (Dia Actions P)) ∧
    (∀N X Z P.
       (∀fs W.
          (fs = frees P) ∧ ¬(X ∉ fs ∨ (X = Z)) ∧ Z ∈ frees N ∧
          (W = variant (frees N ∪ fs) Z) ⇒
          P′ N X (l_Sub [(Z,W)] P)) ∧
       (∀fs.
          (fs = frees P) ∧ ¬(X ∉ fs ∨ (X = Z)) ∧ Z ∉ frees N ⇒
          P′ N X P) ⇒
       P′ N X (nu Z P)) ∧
    (∀N X Z P.
       (∀fs W.
          (fs = frees P) ∧ ¬(X ∉ fs ∨ (X = Z)) ∧ Z ∈ frees N ∧
          (W = variant (frees N ∪ fs) Z) ⇒
          P′ N X (l_Sub [(Z,W)] P)) ∧
       (∀fs.
          (fs = frees P) ∧ ¬(X ∉ fs ∨ (X = Z)) ∧ Z ∉ frees N ⇒
          P′ N X P) ⇒
       P′ N X (mu Z P)) ⇒
    ∀v v1 v2. P′ v v1 v2
```

[Subst_l_Sub]

$\vdash \forall f\ X\ Y.$
    INFINITE $\mathcal{U}$(:'b) $\Rightarrow$ (Subst (propmm Y) X f = l_Sub [(X,Y)] f)

[Subst_LEM]

$\vdash \forall$ *Trans* *Fm* p Z V.
    INFINITE $\mathcal{U}$(:'b) $\Rightarrow$
    (setsat *Trans* (Subst p Z *Fm*) V =
     setsat *Trans* *Fm* (extend_env Z (setsat *Trans* p V) V))

[Subst_not_free]

$\vdash \forall N\ X$ *Fm*. INFINITE $\mathcal{U}$(:'b) $\Rightarrow$ $X \notin$ frees *Fm* $\Rightarrow$ (Subst N X *Fm* = *Fm*)

[Subst_same_size]

$\vdash \forall$ *Fm* X Z.
    INFINITE $\mathcal{U}$(:'b) $\Rightarrow$
    (fmla_size (Subst (propmm X) Z *Fm*) = fmla_size *Fm*)

[uneq_extensions_commute]

$\vdash \forall v\ w\ x\ y\ f.$
    $y \neq x \Rightarrow$
    (extend_env y w (extend_env x v f) =
     extend_env x v (extend_env y w f))

[uneq_mmUpdates_commute]

$\vdash \forall v\ w\ x\ y\ f.$
    $y \neq x \Rightarrow$
    (mmUpdate y (mmUpdate x f v) w =
     mmUpdate x (mmUpdate y f w) v)

[unfold_mu_LEM]

$\vdash \forall$ *Trans* E V Z f.
    INFINITE $\mathcal{U}$(:'b) $\Rightarrow$
    ((E,*Trans*,V) mmsat mu Z f $\iff$
     (E,*Trans*,V) mmsat Subst (mu Z f) Z f)

[unfold_nu_LEM]

$\vdash \forall$ *Trans* E V Z f.
    INFINITE $\mathcal{U}$(:'b) $\Rightarrow$
    ((E,*Trans*,V) mmsat nu Z f $\iff$
     (E,*Trans*,V) mmsat Subst (nu Z f) Z f)

[UNION_SUBSET_MONOTONIC]

$\vdash x_1 \subseteq y_1 \Rightarrow x_2 \subseteq y_2 \Rightarrow x_1 \cup x_2 \subseteq y_1 \cup y_2$

[variant_EXISTS]

$$
\begin{array}{l}
\vdash \exists\, \mathit{variant}.\\
\quad \forall\, \mathit{exclvars}.\\
\quad\quad \mathtt{INFINITE}\ \mathcal{U}(\mathtt{:'variable}) \Rightarrow\\
\quad\quad \mathtt{FINITE}\ \mathit{exclvars} \Rightarrow\\
\quad\quad \forall\, v.\ \mathit{variant}\ \mathit{exclvars}\ v \notin \mathit{exclvars}
\end{array}
$$

[variant_not_in]

$$
\begin{array}{l}
\vdash \forall\, s\ \mathit{excl}.\\
\quad \mathtt{INFINITE}\ \mathcal{U}(\mathtt{:'a}) \Rightarrow \mathtt{FINITE}\ \mathit{excl} \Rightarrow \mathtt{variant}\ \mathit{excl}\ s \notin \mathit{excl}
\end{array}
$$

[vars_finite]

$$\vdash \forall\, f.\ \mathtt{FINITE}\ (\mathtt{vars}\ f)$$

Index