

# An Efficient Normalisation Procedure for Linear Temporal Logic: Isabelle/HOL Formalisation

Salomon Sickert

May 9, 2020

## Abstract

In the mid 80s, Lichtenstein, Pnueli, and Zuck proved a classical theorem stating that every formula of Past LTL (the extension of LTL with past operators) is equivalent to a formula of the form  $\bigwedge_{i=1}^n \mathbf{GF}\varphi_i \vee \mathbf{FG}\psi_i$ , where  $\varphi_i$  and  $\psi_i$  contain only past operators [3, 6]. Some years later, Chang, Manna, and Pnueli built on this result to derive a similar normal form for LTL [2]. Both normalisation procedures have a non-elementary worst-case blow-up, and follow an involved path from formulas to counter-free automata to star-free regular expressions and back to formulas. We improve on both points. We present an executable formalisation of a direct and purely syntactic normalisation procedure for LTL yielding a normal form, comparable to the one by Chang, Manna, and Pnueli, that has only a single exponential blow-up.

## Contents

<b>1</b>	<b>Overview</b>	<b>2</b>
<b>2</b>	<b>A Normal Form for Linear Temporal Logic</b>	<b>2</b>
2.1	LTL Equivalences . . . . .	3
2.2	$\psi[M]_1^{\Pi}$ , $\psi[N]_1^{\Sigma}$ , $\psi[M]_2^{\Sigma}$ , and $\psi[N]_2^{\Pi}$ . . . . .	4
2.3	Main Theorem . . . . .	11
<b>3</b>	<b>Size Bounds</b>	<b>13</b>
3.1	Inequalities and Identities . . . . .	13
3.2	Length . . . . .	13
3.3	Proper Subformulas . . . . .	21
<b>4</b>	<b>Code Export</b>	<b>28</b>

# 1 Overview

This document contains the formalisation of the central results appearing in [4, Sections 4-6]. We refer the interested reader to [4] or to the extended version [5] for an introduction to the topic, related work, intuitive explanations of the proofs, and an application of the normalisation procedure, namely, a translation from LTL to deterministic automata.

The central result of this document is the following theorem:

**Theorem 1.** *Let  $\varphi$  be an LTL formula and let  $\Delta_2$ ,  $\Sigma_1$ ,  $\Sigma_2$ , and  $\Pi_1$  be the classes of LTL formulas from Definition 2. Then  $\varphi$  is equivalent to the following formula from the class  $\Delta_2$ :*

$$\bigvee_{\substack{M \subseteq \mu(\varphi) \\ N \subseteq \nu(\varphi)}} \left( \varphi[M]_2^\Sigma \wedge \bigwedge_{\psi \in M} \mathbf{GF}(\psi[N]_1^\Sigma) \wedge \bigwedge_{\psi \in N} \mathbf{FG}(\psi[M]_1^\Pi) \right)$$

where  $\psi[M]_2^\Sigma$ ,  $\psi[N]_1^\Sigma$ , and  $\psi[M]_1^\Pi$  are functions mapping  $\psi$  to a formula from  $\Sigma_2$ ,  $\Sigma_1$ , and  $\Pi_1$ , respectively.

**Definition 2** (Adapted from [1]). *We define the following classes of LTL formulas:*

- The class  $\Sigma_0 = \Pi_0 = \Delta_0$  is the least set containing all atomic propositions and their negations, and is closed under the application of conjunction and disjunction.
- The class  $\Sigma_{i+1}$  is the least set containing  $\Pi_i$  and is closed under the application of conjunction, disjunction, and the **X**, **U**, and **M** operators.
- The class  $\Pi_{i+1}$  is the least set containing  $\Sigma_i$  and is closed under the application of conjunction, disjunction, and the **X**, **R**, and **W** operators.
- The class  $\Delta_{i+1}$  is the least set containing  $\Sigma_{i+1}$  and  $\Pi_{i+1}$  and is closed under the application of conjunction and disjunction.

## 2 A Normal Form for Linear Temporal Logic

**theory** *Normal-Form* **imports**

*LTL-Master-Theorem.Master-Theorem*

**begin**

## 2.1 LTL Equivalences

Several valid laws of LTL relating strong and weak operators that are useful later.

**lemma** *ltn-strong-weak-2*:

$w \models_n \varphi U_n \psi \iff w \models_n (\varphi \text{ and}_n F_n \psi) W_n \psi$  (**is** *?thesis1*)  
 $w \models_n \varphi M_n \psi \iff w \models_n \varphi R_n (\psi \text{ and}_n F_n \varphi)$  (**is** *?thesis2*)

**proof** –

**have**  $\exists j. \text{suffix } (i + j) w \models_n \psi$   
**if**  $\text{suffix } j w \models_n \psi$  **and**  $\forall j \leq i. \neg \text{suffix } j w \models_n \psi$  **for**  $i j$

**proof**

**from** *that* **have**  $j > i$

**by** (*cases*  $j > i$ ) *auto*

**thus**  $\text{suffix } (i + (j - i)) w \models_n \psi$

**using** *that* **by** *auto*

**qed**

**thus** *?thesis1*

**unfolding** *ltn-strong-weak* **by** *auto*

**next**

**have**  $\exists j. \text{suffix } (i + j) w \models_n \varphi$

**if**  $\text{suffix } j w \models_n \varphi$  **and**  $\forall j < i. \neg \text{suffix } j w \models_n \varphi$  **for**  $i j$

**proof**

**from** *that* **have**  $j \geq i$

**by** (*cases*  $j \geq i$ ) *auto*

**thus**  $\text{suffix } (i + (j - i)) w \models_n \varphi$

**using** *that* **by** *auto*

**qed**

**thus** *?thesis2*

**unfolding** *ltn-strong-weak* **by** *auto*

**qed**

**lemma** *ltn-weak-strong-2*:

$w \models_n \varphi W_n \psi \iff w \models_n \varphi U_n (\psi \text{ or}_n G_n \varphi)$  (**is** *?thesis1*)

$w \models_n \varphi R_n \psi \iff w \models_n (\varphi \text{ or}_n G_n \psi) M_n \psi$  (**is** *?thesis2*)

**proof** –

**have**  $\text{suffix } j w \models_n \varphi$

**if**  $\bigwedge j. j < i \implies \text{suffix } j w \models_n \varphi$  **and**  $\bigwedge j. \text{suffix } (i + j) w \models_n \varphi$  **for**  $i j$

**using** *that(1)[of j]* *that(2)[of j - i]* **by** (*cases*  $j < i$ ) *simp-all*

**thus** *?thesis1*

**unfolding** *ltn-weak-strong* **unfolding** *semantics-ltn.simps* *suffix-suffix*

**by** *blast*

**next**

**have**  $\text{suffix } j w \models_n \psi$

**if**  $\bigwedge j. j \leq i \implies \text{suffix } j w \models_n \psi$  **and**  $\bigwedge j. \text{suffix } (i + j) w \models_n \psi$  **for**  $i j$

**using** *that(1)[of j] that(2)[of j - i]* **by** (*cases j ≤ i*) *simp-all*  
**thus** *?thesis2*  
**unfolding** *ltln-weak-strong unfolding semantics-ltln.simps suffix-suffix*  
**by** *blast*  
**qed**

## 2.2 $\psi[M]_1^\Pi$ , $\psi[N]_1^\Sigma$ , $\psi[M]_2^\Sigma$ , and $\psi[N]_2^\Pi$

The following four functions use "promise sets", named  $M$  or  $N$ , to rewrite arbitrary formulas into formulas from the class  $\Sigma_1$ -,  $\Sigma_2$ -,  $\Pi_1$ -, and  $\Pi_2$ -, respectively. In general the obtained formulas are not equivalent, but under some conditions (as outlined below) they are.

**no-notation** *FG-advice*  $(-[-]_\mu$  [90,60] 89)

**no-notation** *GF-advice*  $(-[-]_\nu$  [90,60] 89)

**notation** *FG-advice*  $(-[-]_{\Sigma_1}$  [90,60] 89)

**notation** *GF-advice*  $(-[-]_{\Pi_1}$  [90,60] 89)

**fun** *flatten-sigma-2*:: *'a ltln ⇒ 'a ltln set ⇒ 'a ltln*  $(-[-]_{\Sigma_2})$

**where**

$(\varphi U_n \psi)[M]_{\Sigma_2} = (\varphi[M]_{\Sigma_2}) U_n (\psi[M]_{\Sigma_2})$   
 $| (\varphi W_n \psi)[M]_{\Sigma_2} = (\varphi[M]_{\Sigma_2}) U_n ((\psi[M]_{\Sigma_2}) \text{or}_n (G_n \varphi[M]_{\Pi_1}))$   
 $| (\varphi M_n \psi)[M]_{\Sigma_2} = (\varphi[M]_{\Sigma_2}) M_n (\psi[M]_{\Sigma_2})$   
 $| (\varphi R_n \psi)[M]_{\Sigma_2} = ((\varphi[M]_{\Sigma_2}) \text{or}_n (G_n \psi[M]_{\Pi_1})) M_n (\psi[M]_{\Sigma_2})$   
 $| (\varphi \text{and}_n \psi)[M]_{\Sigma_2} = (\varphi[M]_{\Sigma_2}) \text{and}_n (\psi[M]_{\Sigma_2})$   
 $| (\varphi \text{or}_n \psi)[M]_{\Sigma_2} = (\varphi[M]_{\Sigma_2}) \text{or}_n (\psi[M]_{\Sigma_2})$   
 $| (X_n \varphi)[M]_{\Sigma_2} = X_n (\varphi[M]_{\Sigma_2})$   
 $| \varphi[M]_{\Sigma_2} = \varphi$

**fun** *flatten-pi-2*:: *'a ltln ⇒ 'a ltln set ⇒ 'a ltln*  $(-[-]_{\Pi_2})$

**where**

$(\varphi W_n \psi)[N]_{\Pi_2} = (\varphi[N]_{\Pi_2}) W_n (\psi[N]_{\Pi_2})$   
 $| (\varphi U_n \psi)[N]_{\Pi_2} = (\varphi[N]_{\Pi_2} \text{and}_n (F_n \psi[N]_{\Sigma_1})) W_n (\psi[N]_{\Pi_2})$   
 $| (\varphi R_n \psi)[N]_{\Pi_2} = (\varphi[N]_{\Pi_2}) R_n (\psi[N]_{\Pi_2})$   
 $| (\varphi M_n \psi)[N]_{\Pi_2} = (\varphi[N]_{\Pi_2}) R_n ((\psi[N]_{\Pi_2}) \text{and}_n (F_n \varphi[N]_{\Sigma_1}))$   
 $| (\varphi \text{and}_n \psi)[N]_{\Pi_2} = (\varphi[N]_{\Pi_2}) \text{and}_n (\psi[N]_{\Pi_2})$   
 $| (\varphi \text{or}_n \psi)[N]_{\Pi_2} = (\varphi[N]_{\Pi_2}) \text{or}_n (\psi[N]_{\Pi_2})$   
 $| (X_n \varphi)[N]_{\Pi_2} = X_n (\varphi[N]_{\Pi_2})$   
 $| \varphi[N]_{\Pi_2} = \varphi$

**lemma** *GF-advice-restriction*:

$\varphi[\mathcal{GF} (\varphi W_n \psi) w]_{\Pi_1} = \varphi[\mathcal{GF} \varphi w]_{\Pi_1}$

$\psi[\mathcal{GF} (\varphi R_n \psi) w]_{\Pi_1} = \psi[\mathcal{GF} \psi w]_{\Pi_1}$

by (metis (no-types, lifting)  $\mathcal{GF}$ -semantics' inf-commute inf-left-commute inf-sup-absorb subformulas $_{\mu}$ .simps(6) GF-advice-inter-subformulas)  
(metis (no-types, lifting) GF-advice-inter  $\mathcal{GF}$ .simps(5)  $\mathcal{GF}$ -semantics'  $\mathcal{GF}$ -subformulas $_{\mu}$  inf.commute sup.boundedE)

**lemma** FG-advice-restriction:

$$\psi[\mathcal{FG} (\varphi U_n \psi) w]_{\Sigma_1} = \psi[\mathcal{FG} \psi w]_{\Sigma_1}$$

$$\varphi[\mathcal{FG} (\varphi M_n \psi) w]_{\Sigma_1} = \varphi[\mathcal{FG} \varphi w]_{\Sigma_1}$$

by (metis (no-types, lifting) FG-advice-inter  $\mathcal{FG}$ .simps(4)  $\mathcal{FG}$ -semantics'  $\mathcal{FG}$ -subformulas $_{\nu}$  inf.commute sup.boundedE)

(metis (no-types, lifting) FG-advice-inter  $\mathcal{FG}$ .simps(7)  $\mathcal{FG}$ -semantics'  $\mathcal{FG}$ -subformulas $_{\nu}$  inf.right-idem inf-commute sup.cobounded1)

**lemma** flatten-sigma-2-intersection:

$$M \cap \text{subformulas}_{\mu} \varphi \subseteq S \implies \varphi[M \cap S]_{\Sigma_2} = \varphi[M]_{\Sigma_2}$$

by (induction  $\varphi$ ) (simp; blast intro: GF-advice-inter)+

**lemma** flatten-sigma-2-intersection-eq:

$$M \cap \text{subformulas}_{\mu} \varphi = M' \implies \varphi[M']_{\Sigma_2} = \varphi[M]_{\Sigma_2}$$

using flatten-sigma-2-intersection by auto

**lemma** flatten-sigma-2-monotone:

$$w \models_n \varphi[M]_{\Sigma_2} \implies M \subseteq M' \implies w \models_n \varphi[M']_{\Sigma_2}$$

by (induction  $\varphi$  arbitrary: w)

(simp; blast dest: GF-advice-monotone)+

**lemma** flatten-pi-2-intersection:

$$N \cap \text{subformulas}_{\nu} \varphi \subseteq S \implies \varphi[N \cap S]_{\Pi_2} = \varphi[N]_{\Pi_2}$$

by (induction  $\varphi$ ) (simp; blast intro: FG-advice-inter)+

**lemma** flatten-pi-2-intersection-eq:

$$N \cap \text{subformulas}_{\nu} \varphi = N' \implies \varphi[N']_{\Pi_2} = \varphi[N]_{\Pi_2}$$

using flatten-pi-2-intersection by auto

**lemma** flatten-pi-2-monotone:

$$w \models_n \varphi[N]_{\Pi_2} \implies N \subseteq N' \implies w \models_n \varphi[N']_{\Pi_2}$$

by (induction  $\varphi$  arbitrary: w)

(simp; blast dest: FG-advice-monotone)+

**lemma** ltl-weak-strong-stable-words-1:

$$w \models_n (\varphi W_n \psi) \iff w \models_n \varphi U_n (\psi \text{ or}_n (G_n \varphi[\mathcal{GF} \varphi w]_{\Pi_1})) \text{ (is ?lhs} \\ \iff \text{?rhs)}$$

**proof**

assume ?lhs

**moreover**

{
  
  **assume** *assm*:  $w \models_n G_n \varphi$ 
  
  **moreover**
  
  **obtain** *i* **where**  $\bigwedge j. \mathcal{F} \varphi (\text{suffix } i \ w) \subseteq \mathcal{GF} \varphi \ w$ 
  
  **by** (*metis MOST-nat-le GF-suffix  $\mu$ -stable-def order-refl suffix- $\mu$ -stable*)
  
  **hence**  $\bigwedge j. \mathcal{F} \varphi (\text{suffix } i \ (\text{suffix } j \ w)) \subseteq \mathcal{GF} \varphi \ w$ 
  
  **by** (*metis F-suffix GF-F-subset GF-suffix semiring-normalization-rules(24)*)
  
*subset-Un-eq suffix-suffix sup.orderE*)
  
  **ultimately**
  
  **have**  $\text{suffix } i \ w \models_n G_n (\varphi[\mathcal{GF} \varphi \ w]_{\Pi 1})$ 
  
  **using** *GF-advice-a1[OF  $\bigwedge j. \mathcal{F} \varphi (\text{suffix } i \ (\text{suffix } j \ w)) \subseteq \mathcal{GF} \varphi \ w$ ]*
  
  **by** (*simp add: add commute*)
  
  **hence** *?rhs*
  
  **using** *assm* **by** *auto*
  
}

**moreover**

**have**  $w \models_n \varphi \ U_n \ \psi \implies \text{?rhs}$ 
  
**by** *auto*

**ultimately**

**show** *?rhs*
  
  **using** *ltln-weak-to-strong(1)* **by** *blast*

**next**
  
**assume** *?rhs*
  
**thus** *?lhs*
  
  **unfolding** *ltln-weak-strong-2* **unfolding** *semantics-ltn.simps*
  
  **by** (*metis GF-suffix order-refl GF-advice-a2*)

**qed**

**lemma** *ltln-weak-strong-stable-words-2*:
  
 $w \models_n (\varphi \ R_n \ \psi) \longleftrightarrow w \models_n (\varphi \ or_n \ (G_n \ \psi[\mathcal{GF} \ \psi \ w]_{\Pi 1})) \ M_n \ \psi$  (**is** *?lhs*
  
 $\longleftrightarrow \text{?rhs}$ )

**proof**
  
  **assume** *?lhs*

**moreover**

{

**assume** *assm*:  $w \models_n G_n \psi$   
**moreover**  
**obtain**  $i$  **where**  $\bigwedge j. \mathcal{F} \psi (\text{suffix } i \ w) \subseteq \mathcal{GF} \psi \ w$   
**by** (*metis MOST-nat-le GF-suffix  $\mu$ -stable-def order-refl suffix- $\mu$ -stable*)  
**hence**  $\bigwedge j. \mathcal{F} \psi (\text{suffix } i \ (\text{suffix } j \ w)) \subseteq \mathcal{GF} \psi \ w$   
**by** (*metis F-suffix GF-F-subset GF-suffix semiring-normalization-rules(24)*  
*subset-Un-eq suffix-suffix sup.orderE*)  
**ultimately**  
**have**  $\text{suffix } i \ w \models_n G_n (\psi[\mathcal{GF} \psi \ w]_{\Pi 1})$   
**using** *GF-advice-a1[OF  $\bigwedge j. \mathcal{F} \psi (\text{suffix } i \ (\text{suffix } j \ w)) \subseteq \mathcal{GF} \psi \ w$ ]*  
**by** (*simp add: add commute*)  
**hence** *?rhs*  
**using** *assm* **by** *auto*  
**}**

**moreover**

**have**  $w \models_n \varphi \ M_n \psi \implies ?rhs$   
**by** *auto*

**ultimately**

**show** *?rhs*  
**using** *ltln-weak-to-strong* **by** *blast*

**next**

**assume** *?rhs*  
**thus** *?lhs*  
**unfolding** *ltln-weak-strong-2* **unfolding** *semantics-ltn.simps*  
**by** (*metis GF-advice-a2 GF-suffix order-refl*)

**qed**

**lemma** *ltln-weak-strong-stable-words*:

$w \models_n (\varphi \ W_n \ \psi) \longleftrightarrow w \models_n \varphi \ U_n (\psi \ \text{or}_n (G_n \ \varphi[\mathcal{GF} (\varphi \ W_n \ \psi) \ w]_{\Pi 1}))$   
 $w \models_n (\varphi \ R_n \ \psi) \longleftrightarrow w \models_n (\varphi \ \text{or}_n (G_n \ \psi[\mathcal{GF} (\varphi \ R_n \ \psi) \ w]_{\Pi 1})) \ M_n \ \psi$   
**unfolding** *ltln-weak-strong-stable-words-1* *ltln-weak-strong-stable-words-2*  
*GF-advice-restriction* **by** *simp+*

**lemma** *flatten-sigma-2-IH-lifting*:

**assumes**  $\psi \in \text{subfrmlsn } \varphi$   
**assumes**  $\text{suffix } i \ w \models_n \psi[\mathcal{GF} \psi (\text{suffix } i \ w)]_{\Sigma 2} = \text{suffix } i \ w \models_n \psi$   
**shows**  $\text{suffix } i \ w \models_n \psi[\mathcal{GF} \varphi \ w]_{\Sigma 2} = \text{suffix } i \ w \models_n \psi$   
**by** (*metis (no-types, lifting) inf.absorb-iff2 inf-assoc inf-commute assms(2)*  
*GF-suffix flatten-sigma-2-intersection-eq[of  $\mathcal{GF} \varphi \ w \ \psi \ \mathcal{GF} \psi \ w$ ] GF-semantics'*  
*subformulas $_{\mu}$ -subset[OF assms(1)]*)

**lemma** *flatten-sigma-2-correct*:  
 $w \models_n \varphi[\mathcal{GF} \varphi w]_{\Sigma 2} \longleftrightarrow w \models_n \varphi$   
**proof** (*induction  $\varphi$  arbitrary:  $w$* )  
  **case** (*And-ltln  $\varphi 1 \varphi 2$* )  
  **then show** *?case*  
    **using** *flatten-sigma-2-IH-lifting[ $of - \varphi 1$  and $_n \varphi 2 0$ ]* **by** *simp*  
**next**  
  **case** (*Or-ltln  $\varphi 1 \varphi 2$* )  
  **then show** *?case*  
    **using** *flatten-sigma-2-IH-lifting[ $of - \varphi 1$  or $_n \varphi 2 0$ ]* **by** *simp*  
**next**  
  **case** (*Next-ltln  $\varphi$* )  
  **then show** *?case*  
    **using** *flatten-sigma-2-IH-lifting[ $of - X_n \varphi 1$ ]* **by** *fastforce*  
**next**  
  **case** (*Until-ltln  $\varphi 1 \varphi 2$* )  
  **then show** *?case*  
    **using** *flatten-sigma-2-IH-lifting[ $of - \varphi 1 U_n \varphi 2$ ]* **by** *fastforce*  
**next**  
  **case** (*Release-ltln  $\varphi 1 \varphi 2$* )  
  **then show** *?case*  
    **unfolding** *ltln-weak-strong-stable-words*  
    **using** *flatten-sigma-2-IH-lifting[ $of - \varphi 1 R_n \varphi 2$ ]* **by** *fastforce*  
**next**  
  **case** (*WeakUntil-ltln  $\varphi 1 \varphi 2$* )  
  **then show** *?case*  
    **unfolding** *ltln-weak-strong-stable-words*  
    **using** *flatten-sigma-2-IH-lifting[ $of - \varphi 1 W_n \varphi 2$ ]* **by** *fastforce*  
**next**  
  **case** (*StrongRelease-ltln  $\varphi 1 \varphi 2$* )  
  **then show** *?case*  
    **using** *flatten-sigma-2-IH-lifting[ $of - \varphi 1 M_n \varphi 2$ ]* **by** *fastforce*  
**qed** *auto*

**lemma** *ltln-strong-weak-stable-words-1*:  
 $w \models_n \varphi U_n \psi \longleftrightarrow w \models_n (\varphi \text{ and}_n (F_n \psi[\mathcal{FG} \psi w]_{\Sigma 1})) W_n \psi$  (**is** *?lhs*  
 $\longleftrightarrow$  *?rhs*)  
**proof**  
  **assume** *?rhs*  
  
  **moreover**  
  
  **obtain** *i* **where**  $\nu$ -*stable  $\psi$  (suffix  $i w$ )*



by (*metis MOST-nat less-Suc-eq suffix- $\nu$ -stable*)  
 hence  $\forall \psi \in \mathcal{FG} \ \psi \ w. \text{ suffix } i \ w \models_n G_n \ \psi$   
 using *FG-suffix G-elim  $\nu$ -stable-def* by *blast*

{  
   assume *assm*:  $w \models_n G_n (\varphi \text{ and}_n (F_n \ \psi [\mathcal{FG} \ \psi \ w]_{\Sigma 1}))$   
   hence  $\text{suffix } i \ w \models_n (F_n \ \psi) [\mathcal{FG} \ \psi \ w]_{\Sigma 1}$   
     by *simp*  
   hence  $\text{suffix } i \ w \models_n F_n \ \psi$   
     by (*blast dest: FG-advice-b2-helper*[*OF*  $\forall \psi \in \mathcal{FG} \ \psi \ w. \text{ suffix } i \ w \models_n G_n \ \psi$ ])  
   hence  $w \models_n \varphi \ U_n \ \psi$   
     using *assm* by *auto*  
 }

ultimately

show *?lhs*  
 by (*meson ltln-weak-to-strong*(1) *semantics-ltln.simps*(5) *until-and-left-distrib*)

next

assume *?lhs*

moreover

have  $\bigwedge i. \text{ suffix } i \ w \models_n \psi \implies \text{ suffix } i \ w \models_n \psi [\mathcal{FG} \ \psi \ w]_{\Sigma 1}$   
 using *FG-suffix* by (*blast intro: FG-advice-b1*)

ultimately

show *?rhs*  
 unfolding *ltln-strong-weak-2* by *fastforce*

qed

**lemma** *ltln-strong-weak-stable-words-2*:  
 $w \models_n \varphi \ M_n \ \psi \longleftrightarrow w \models_n \varphi \ R_n (\psi \text{ and}_n (F_n \ \varphi [\mathcal{FG} \ \varphi \ w]_{\Sigma 1}))$  (**is** *?lhs*  $\longleftrightarrow$  *?rhs*)

**proof**

assume *?rhs*

moreover

obtain *i* where  $\nu$ -stable  $\varphi$  (*suffix* *i* *w*)  
 by (*metis MOST-nat less-Suc-eq suffix- $\nu$ -stable*)  
 hence  $\forall \psi \in \mathcal{FG} \ \varphi \ w. \text{ suffix } i \ w \models_n G_n \ \psi$

**using** *FG-suffix G-elim ν-stable-def* **by** *blast*

{
  
**assume** *assm*:  $w \models_n G_n (\psi \text{ and}_n (F_n \varphi [\mathcal{FG} \varphi w]_{\Sigma 1}))$ 
  
**hence** *suffix i w*  $\models_n (F_n \varphi) [\mathcal{FG} \varphi w]_{\Sigma 1}$ 
  
**by** *simp*
  
**hence** *suffix i w*  $\models_n F_n \varphi$ 
  
**by** (*blast dest: FG-advice-b2-helper*[*OF*  $\forall \psi \in \mathcal{FG} \varphi w. \text{suffix } i \ w \models_n G_n \psi$ ])
  
**hence**  $w \models_n \varphi M_n \psi$ 
  
**using** *assm* **by** *auto*
  
}

**ultimately**

**show** *?lhs*
  
**using** *ltln-weak-to-strong(3) semantics-ltln.simps(5) strong-release-and-right-distrib*
  
**by** *blast*

**next**
  
**assume** *?lhs*

**moreover**

**have**  $\bigwedge i. \text{suffix } i \ w \models_n \varphi \implies \text{suffix } i \ w \models_n \varphi [\mathcal{FG} \varphi w]_{\Sigma 1}$ 
  
**using** *FG-suffix* **by** (*blast intro: FG-advice-b1*)

**ultimately**

**show** *?rhs*
  
**unfolding** *ltln-strong-weak-2* **by** *fastforce*

**qed**

**lemma** *ltln-strong-weak-stable-words*:
  
 $w \models_n \varphi U_n \psi \iff w \models_n (\varphi \text{ and}_n (F_n \psi [\mathcal{FG} (\varphi U_n \psi) w]_{\Sigma 1})) W_n \psi$ 
  
 $w \models_n \varphi M_n \psi \iff w \models_n \varphi R_n (\psi \text{ and}_n (F_n \varphi [\mathcal{FG} (\varphi M_n \psi) w]_{\Sigma 1}))$ 
  
**unfolding** *ltln-strong-weak-stable-words-1 ltln-strong-weak-stable-words-2*
  
*FG-advice-restriction* **by** *simp+*

**lemma** *flatten-pi-2-IH-lifting*:
  
**assumes**  $\psi \in \text{subfrmlsn } \varphi$ 
  
**assumes**  $\text{suffix } i \ w \models_n \psi [\mathcal{FG} \psi (\text{suffix } i \ w)]_{\Pi 2} = \text{suffix } i \ w \models_n \psi$ 
  
**shows**  $\text{suffix } i \ w \models_n \psi [\mathcal{FG} \varphi w]_{\Pi 2} = \text{suffix } i \ w \models_n \psi$ 
  
**by** (*metis (no-types, lifting) inf.absorb-iff2 inf-assoc inf-commute assms(2)*
  
*FG-suffix flatten-pi-2-intersection-eq*[*of*  $\mathcal{FG} \varphi w \psi \mathcal{FG} \psi w$ ] *FG-semantics'*)

*subformulas<sub>ν</sub>-subset*[*OF assms*(1)]

**lemma** *flatten-pi-2-correct*:

$w \models_n \varphi[\mathcal{FG} \varphi w]_{\Pi_2} \longleftrightarrow w \models_n \varphi$

**proof** (*induction*  $\varphi$  *arbitrary*:  $w$ )

**case** (*And-ltln*  $\varphi_1 \varphi_2$ )

**then show** *?case*

**using** *flatten-pi-2-IH-lifting*[*of* -  $\varphi_1$  *and* <sub>$n$</sub>   $\varphi_2$  0] **by** *simp*

**next**

**case** (*Or-ltln*  $\varphi_1 \varphi_2$ )

**then show** *?case*

**using** *flatten-pi-2-IH-lifting*[*of* -  $\varphi_1$  *or* <sub>$n$</sub>   $\varphi_2$  0] **by** *simp*

**next**

**case** (*Next-ltln*  $\varphi$ )

**then show** *?case*

**using** *flatten-pi-2-IH-lifting*[*of* -  $X_n \varphi$  1] **by** *fastforce*

**next**

**case** (*Until-ltln*  $\varphi_1 \varphi_2$ )

**then show** *?case*

**unfolding** *ltln-strong-weak-stable-words*

**using** *flatten-pi-2-IH-lifting*[*of* -  $\varphi_1 U_n \varphi_2$ ] **by** *fastforce*

**next**

**case** (*Release-ltln*  $\varphi_1 \varphi_2$ )

**then show** *?case*

**using** *flatten-pi-2-IH-lifting*[*of* -  $\varphi_1 R_n \varphi_2$ ] **by** *fastforce*

**next**

**case** (*WeakUntil-ltln*  $\varphi_1 \varphi_2$ )

**then show** *?case*

**using** *flatten-pi-2-IH-lifting*[*of* -  $\varphi_1 W_n \varphi_2$ ] **by** *fastforce*

**next**

**case** (*StrongRelease-ltln*  $\varphi_1 \varphi_2$ )

**then show** *?case*

**unfolding** *ltln-strong-weak-stable-words*

**using** *flatten-pi-2-IH-lifting*[*of* -  $\varphi_1 M_n \varphi_2$ ] **by** *fastforce*

**qed** *auto*

## 2.3 Main Theorem

Using the four previously defined functions we obtain our normal form.

**theorem** *normal-form-with-flatten-sigma-2*:

$w \models_n \varphi \longleftrightarrow$

$(\exists M \subseteq \text{subformulas}_\mu \varphi. \exists N \subseteq \text{subformulas}_\nu \varphi.$

$w \models_n \varphi[M]_{\Sigma_2} \wedge (\forall \psi \in M. w \models_n G_n (F_n \psi[N]_{\Sigma_1})) \wedge (\forall \psi \in N. w$

$\models_n F_n (G_n \psi[M]_{\Pi 1}))$  (is ?lhs  $\longleftrightarrow$  ?rhs)  
**proof**  
 assume ?lhs  
 then have  $w \models_n \varphi[\mathcal{GF} \varphi w]_{\Sigma 2}$   
 using *flatten-sigma-2-correct* by *blast*  
 then show ?rhs  
 using  *$\mathcal{GF}$ -subformulas $_{\mu}$*   *$\mathcal{FG}$ -subformulas $_{\nu}$*   *$\mathcal{GF}$ -implies- $\mathcal{GF}$*   *$\mathcal{FG}$ -implies- $\mathcal{FG}$*   
**by metis**  
**next**  
 assume ?rhs  
 then obtain  $M N$  where  $w \models_n \varphi[M]_{\Sigma 2}$  and  $M \subseteq \mathcal{GF} \varphi w$  and  $N \subseteq \mathcal{FG} \varphi w$   
 using *X- $\mathcal{GF}$ -Y- $\mathcal{FG}$*  by *blast*  
 then have  $w \models_n \varphi[\mathcal{GF} \varphi w]_{\Sigma 2}$   
 using *flatten-sigma-2-monotone* by *blast*  
 then show ?lhs  
 using *flatten-sigma-2-correct* by *blast*  
**qed**

**theorem** *normal-form-with-flatten-pi-2:*

$w \models_n \varphi \longleftrightarrow$   
 $(\exists M \subseteq \text{subformulas}_{\mu} \varphi. \exists N \subseteq \text{subformulas}_{\nu} \varphi.$   
 $w \models_n \varphi[N]_{\Pi 2} \wedge (\forall \psi \in M. w \models_n G_n (F_n \psi[N]_{\Sigma 1})) \wedge (\forall \psi \in N. w \models_n$   
 $F_n (G_n \psi[M]_{\Pi 1}))$ ) (is ?lhs  $\longleftrightarrow$  ?rhs)

**proof**  
 assume ?lhs  
 then have  $w \models_n \varphi[\mathcal{FG} \varphi w]_{\Pi 2}$   
 using *flatten-pi-2-correct* by *blast*  
 then show ?rhs  
 using  *$\mathcal{GF}$ -subformulas $_{\mu}$*   *$\mathcal{FG}$ -subformulas $_{\nu}$*   *$\mathcal{GF}$ -implies- $\mathcal{GF}$*   *$\mathcal{FG}$ -implies- $\mathcal{FG}$*   
**by metis**  
**next**  
 assume ?rhs  
 then obtain  $M N$  where  $w \models_n \varphi[N]_{\Pi 2}$  and  $M \subseteq \mathcal{GF} \varphi w$  and  $N \subseteq \mathcal{FG} \varphi w$   
 using *X- $\mathcal{GF}$ -Y- $\mathcal{FG}$*  by *metis*  
 then have  $w \models_n \varphi[\mathcal{FG} \varphi w]_{\Pi 2}$   
 using *flatten-pi-2-monotone* by *metis*  
 then show ?lhs  
 using *flatten-pi-2-correct* by *blast*  
**qed**

**end**

### 3 Size Bounds

We prove an exponential upper bound for the normalisation procedure. Moreover, we show that the number of proper subformulas, which correspond to states very-weak alternating automata (A1W), is only linear for each disjunct.

**theory** *Normal-Form-Complexity* **imports**

*Normal-Form*

**begin**

#### 3.1 Inequalities and Identities

**lemma** *inequality-1*:

$y > 0 \implies y + 3 \leq (2 :: \text{nat}) \wedge (y + 1)$

**by** (*induction y*) (*simp, fastforce*)

**lemma** *inequality-2*:

$x > 0 \implies y > 0 \implies ((2 :: \text{nat}) \wedge (x + 1)) + (2 \wedge (y + 1)) \leq (2 \wedge (x + y + 1))$

**by** (*induction x; simp; induction y; simp; fastforce*)

**lemma** *size-gr-0*:

$\text{size } (\varphi :: 'a \text{ ltn}) > 0$

**by** (*cases*  $\varphi$ ) *simp-all*

**lemma** *sum-associative*:

$\text{finite } X \implies (\sum x \in X. f x + c) = (\sum x \in X. f x) + \text{card } X * c$

**by** (*induction rule: finite-induct*) *simp-all*

#### 3.2 Length

We prove that the length (size) of the resulting formula in normal form is at most exponential.

**lemma** *flatten-sigma-1-length*:

$\text{size } (\varphi[N]_{\Sigma 1}) \leq \text{size } \varphi$

**by** (*induction*  $\varphi$ ) *simp-all*

**lemma** *flatten-pi-1-length*:

$\text{size } (\varphi[M]_{\Pi 1}) \leq \text{size } \varphi$

**by** (*induction*  $\varphi$ ) *simp-all*

**lemma** *flatten-sigma-2-length*:

$\text{size } (\varphi[M]_{\Sigma 2}) \leq 2 \wedge (\text{size } \varphi + 1)$

```

proof (induction  $\varphi$ )
  case (And-ltln  $\varphi1$   $\varphi2$ )
    hence  $\text{size } (\varphi1 \text{ and}_n \varphi2)[M]_{\Sigma2} \leq (2 \wedge (\text{size } \varphi1 + 1)) + (2 \wedge (\text{size } \varphi2 + 1)) + 1$ 
    by simp
    also
    have  $\dots \leq 2 \wedge (\text{size } \varphi1 + \text{size } \varphi2 + 1) + 1$ 
    using inequality-2[OF size-gr-0 size-gr-0] by simp
    also
    have  $\dots \leq 2 \wedge (\text{size } (\varphi1 \text{ and}_n \varphi2) + 1)$ 
    by simp
    finally
    show ?case.
next
  case (Or-ltln  $\varphi1$   $\varphi2$ )
    hence  $\text{size } (\varphi1 \text{ or}_n \varphi2)[M]_{\Sigma2} \leq (2 \wedge (\text{size } \varphi1 + 1)) + (2 \wedge (\text{size } \varphi2 + 1)) + 1$ 
    by simp
    also
    have  $\dots \leq 2 \wedge (\text{size } \varphi1 + \text{size } \varphi2 + 1) + 1$ 
    using inequality-2[OF size-gr-0 size-gr-0] by simp
    also
    have  $\dots \leq 2 \wedge (\text{size } (\varphi1 \text{ or}_n \varphi2) + 1)$ 
    by simp
    finally
    show ?case.
next
  case (Next-ltln  $\varphi$ )
  then show ?case
    using le-Suc-eq by fastforce
next
  case (WeakUntil-ltln  $\varphi1$   $\varphi2$ )
    hence  $\text{size } (\varphi1 \text{ W}_n \varphi2)[M]_{\Sigma2} \leq 2 \wedge (\text{size } \varphi1 + 1) + 2 \wedge (\text{size } \varphi2 + 1) + \text{size } \varphi1 + 4$ 
    by (simp, simp add: add.commute add-mono flatten-pi-1-length)
    also
    have  $\dots \leq 2 \wedge (\text{size } \varphi2 + 1) + 2 * 2 \wedge (\text{size } \varphi1 + 1) + 1$ 
    using inequality-1[OF size-gr-0, of  $\varphi1$ ] by simp
    also
    have  $\dots \leq 2 * (2 \wedge (\text{size } \varphi1 + 1) + 2 \wedge (\text{size } \varphi2 + 1))$ 
    by simp
    also
    have  $\dots \leq 2 * 2 \wedge (\text{size } \varphi1 + \text{size } \varphi2 + 1)$ 
    using inequality-2[OF size-gr-0 size-gr-0] mult-le-mono2 by blast

```

```

also
have ... =  $2^{\text{size } (\varphi1 \ W_n \ \varphi2) + 1}$ 
  by simp
finally
show ?case.
next
  case (StrongRelease-ltln  $\varphi1 \ \varphi2$ )
  hence  $\text{size } (\varphi1 \ M_n \ \varphi2)[M]_{\Sigma 2} \leq (2^{\text{size } \varphi1 + 1}) + (2^{\text{size } \varphi2 + 1}) + 1$ 
  by simp
  also
  have ...  $\leq 2^{\text{size } \varphi1 + \text{size } \varphi2 + 1} + 1$ 
    using inequality-2[OF size-gr-0 size-gr-0] by simp
  also
  have ...  $\leq 2^{\text{size } (\varphi1 \ M_n \ \varphi2) + 1}$ 
    by simp
  finally
  show ?case.
next
  case (Until-ltln  $\varphi1 \ \varphi2$ )
  hence  $\text{size } (\varphi1 \ U_n \ \varphi2)[M]_{\Sigma 2} \leq (2^{\text{size } \varphi1 + 1}) + (2^{\text{size } \varphi2 + 1}) + 1$ 
  by simp
  also
  have ...  $\leq 2^{\text{size } \varphi1 + \text{size } \varphi2 + 1} + 1$ 
    using inequality-2[OF size-gr-0 size-gr-0] by simp
  also
  have ...  $\leq 2^{\text{size } (\varphi1 \ U_n \ \varphi2) + 1}$ 
    by simp
  finally
  show ?case.
next
  case (Release-ltln  $\varphi1 \ \varphi2$ )
  hence  $\text{size } (\varphi1 \ R_n \ \varphi2)[M]_{\Sigma 2} \leq 2^{\text{size } \varphi1 + 1} + 2^{\text{size } \varphi2 + 1} + \text{size } \varphi2 + 4$ 
  by (simp, simp add: add commute add-mono flatten-pi-1-length)
  also
  have ...  $\leq 2^{\text{size } \varphi1 + 1} + 2 * 2^{\text{size } \varphi2 + 1} + 1$ 
    using inequality-1[OF size-gr-0, of  $\varphi2$ ] by simp
  also
  have ...  $\leq 2 * (2^{\text{size } \varphi1 + 1} + 2^{\text{size } \varphi2 + 1})$ 
    by simp
  also
  have ...  $\leq 2 * 2^{\text{size } \varphi1 + \text{size } \varphi2 + 1}$ 

```

```

    using inequality-2[OF size-gr-0 size-gr-0] mult-le-mono2 by blast
  also
  have ... = 2 ^ (size (φ1 Rn φ2) + 1)
    by simp
  finally
  show ?case .
qed auto

lemma flatten-pi-2-length:
  size (φ[N]Π2) ≤ 2 ^ (size φ + 1)
proof (induction φ)
  case (And-ltln φ1 φ2)
  hence size (φ1 andn φ2)[N]Π2 ≤ (2 ^ (size φ1 + 1)) + (2 ^ (size φ2
+ 1)) + 1
    by simp
  also
  have ... ≤ 2 ^ (size φ1 + size φ2 + 1) + 1
    using inequality-2[OF size-gr-0 size-gr-0] by simp
  also
  have ... ≤ 2 ^ (size (φ1 andn φ2) + 1)
    by simp
  finally
  show ?case.
next
  case (Or-ltln φ1 φ2)
  hence size (φ1 orn φ2)[N]Π2 ≤ (2 ^ (size φ1 + 1)) + (2 ^ (size φ2 +
1)) + 1
    by simp
  also
  have ... ≤ 2 ^ (size φ1 + size φ2 + 1) + 1
    using inequality-2[OF size-gr-0 size-gr-0] by simp
  also
  have ... ≤ 2 ^ (size (φ1 orn φ2) + 1)
    by simp
  finally
  show ?case.
next
  case (Next-ltln φ)
  then show ?case
    using le-Suc-eq by fastforce
next
  case (Until-ltln φ1 φ2)
  hence size (φ1 Un φ2)[N]Π2 ≤ 2 ^ (size φ1 + 1) + 2 ^ (size φ2 + 1)
+ size φ2 + 4

```



```

    by (simp, simp add: add.commute add-mono flatten-sigma-1-length)
  also
  have ... ≤ 2 ^ (size φ1 + 1) + 2 * 2 ^ (size φ2 + 1) + 1
    using inequality-1[OF size-gr-0, of φ2] by simp
  also
  have ... ≤ 2 * (2 ^ (size φ1 + 1) + 2 ^ (size φ2 + 1))
    by simp
  also
  have ... ≤ 2 * 2 ^ (size φ1 + size φ2 + 1)
    using inequality-2[OF size-gr-0 size-gr-0] mult-le-mono2 by blast
  also
  have ... = 2 ^ (size (φ1 Un φ2) + 1)
    by simp
  finally
  show ?case.
next
case (Release-ltln φ1 φ2)
hence size (φ1 Rn φ2)[N]Π2 ≤ (2 ^ (size φ1 + 1)) + (2 ^ (size φ2 +
1)) + 1
  by simp
also
have ... ≤ 2 ^ (size φ1 + size φ2 + 1) + 1
  using inequality-2[OF size-gr-0 size-gr-0] by simp
also
have ... ≤ 2 ^ (size (φ1 Rn φ2) + 1)
  by simp
finally
show ?case.
next
case (WeakUntil-ltln φ1 φ2)
hence size (φ1 Wn φ2)[N]Π2 ≤ (2 ^ (size φ1 + 1)) + (2 ^ (size φ2 +
1)) + 1
  by simp
also
have ... ≤ 2 ^ (size φ1 + size φ2 + 1) + 1
  using inequality-2[OF size-gr-0 size-gr-0] by simp
also
have ... ≤ 2 ^ (size (φ1 Wn φ2) + 1)
  by simp
finally
show ?case.
next
case (StrongRelease-ltln φ1 φ2)
hence size (φ1 Mn φ2)[N]Π2 ≤ 2 ^ (size φ1 + 1) + 2 ^ (size φ2 + 1)

```

$+ \text{size } \varphi 1 + 4$   
**by** (*simp*, *simp add: add.commute add-mono flatten-sigma-1-length*)  
**also**  
**have**  $\dots \leq 2 \wedge (\text{size } \varphi 2 + 1) + 2 * 2 \wedge (\text{size } \varphi 1 + 1) + 1$   
**using** *inequality-1[OF size-gr-0, of  $\varphi 1$ ]* **by** *simp*  
**also**  
**have**  $\dots \leq 2 * (2 \wedge (\text{size } \varphi 1 + 1) + 2 \wedge (\text{size } \varphi 2 + 1))$   
**by** *simp*  
**also**  
**have**  $\dots \leq 2 * 2 \wedge (\text{size } \varphi 1 + \text{size } \varphi 2 + 1)$   
**using** *inequality-2[OF size-gr-0 size-gr-0] mult-le-mono2* **by** *blast*  
**also**  
**have**  $\dots = 2 \wedge (\text{size } (\varphi 1 M_n \varphi 2) + 1)$   
**by** *simp*  
**finally**  
**show** *?case* .  
**qed** *auto*

**definition** *normal-form-length-upper-bound*  
**where** *normal-form-length-upper-bound*  $\varphi$   
 $\equiv (2 :: \text{nat}) \wedge (\text{size } \varphi) * (2 \wedge (\text{size } \varphi + 1) + 2 * (\text{size } \varphi + 2) \wedge 2)$

**definition** *normal-form-disjunct-with-flatten-pi-2-length*  
**where** *normal-form-disjunct-with-flatten-pi-2-length*  $\varphi M N$   
 $\equiv \text{size } (\varphi[N]_{\Pi 2}) + (\sum \psi \in M. \text{size } (\psi[N]_{\Sigma 1}) + 2) + (\sum \psi \in N. \text{size } (\psi[M]_{\Pi 1}) + 2)$

**definition** *normal-form-with-flatten-pi-2-length*  
**where** *normal-form-with-flatten-pi-2-length*  $\varphi$   
 $\equiv \sum (M, N) \in \{(M, N) \mid M N. M \subseteq \text{subformulas}_\mu \varphi \wedge N \subseteq \text{subformulas}_\nu \varphi\}$ . *normal-form-disjunct-with-flatten-pi-2-length*  $\varphi M N$

**definition** *normal-form-disjunct-with-flatten-sigma-2-length*  
**where** *normal-form-disjunct-with-flatten-sigma-2-length*  $\varphi M N$   
 $\equiv \text{size } (\varphi[M]_{\Sigma 2}) + (\sum \psi \in M. \text{size } (\psi[N]_{\Sigma 1}) + 2) + (\sum \psi \in N. \text{size } (\psi[M]_{\Pi 1}) + 2)$

**definition** *normal-form-with-flatten-sigma-2-length*  
**where** *normal-form-with-flatten-sigma-2-length*  $\varphi$   
 $\equiv \sum (M, N) \in \{(M, N) \mid M N. M \subseteq \text{subformulas}_\mu \varphi \wedge N \subseteq \text{subformulas}_\nu \varphi\}$ . *normal-form-disjunct-with-flatten-sigma-2-length*  $\varphi M N$

**lemma** *normal-form-disjunct-length-upper-bound:*  
**assumes**

$M \subseteq \text{subformulas}_\mu \varphi$

$N \subseteq \text{subformulas}_\nu \varphi$

**shows**

*normal-form-disjunct-with-flatten-sigma-2-length*  $\varphi$   $M N \leq 2 \wedge (\text{size } \varphi + 1) + 2 * (\text{size } \varphi + 2) \wedge 2$  (**is** *?thesis1*)

*normal-form-disjunct-with-flatten-pi-2-length*  $\varphi$   $M N \leq 2 \wedge (\text{size } \varphi + 1) + 2 * (\text{size } \varphi + 2) \wedge 2$  (**is** *?thesis2*)

**proof** –

**let**  $?n = \text{size } \varphi$

**let**  $?b = 2 \wedge (?n + 1) + ?n * (?n + 2) + ?n * (?n + 2)$

**have** *finite-M*: *finite*  $M$  **and** *card-M*:  $\text{card } M \leq ?n$

**by** (*metis* *assms(1)* *finite-subset* *subformulas<sub>μ</sub>-finite*)

(*meson* *assms(1)* *card-mono* *order-trans* *subformulas<sub>μ</sub>-subfrmlsn* *subfrmlsn-card* *subfrmlsn-finite*)

**have** *finite-N*: *finite*  $N$  **and** *card-N*:  $\text{card } N \leq ?n$

**by** (*metis* *assms(2)* *finite-subset* *subformulas<sub>ν</sub>-finite*)

(*meson* *assms(2)* *card-mono* *order-trans* *subformulas<sub>ν</sub>-subfrmlsn* *subfrmlsn-card* *subfrmlsn-finite*)

**have** *size-M*:  $\bigwedge \psi. \psi \in M \implies \text{size } \psi \leq \text{size } \varphi$

**and** *size-N*:  $\bigwedge \psi. \psi \in N \implies \text{size } \psi \leq \text{size } \varphi$

**by** (*metis* *assms(1)* *eq-iff* *in-mono* *less-imp-le* *subformulas<sub>μ</sub>-subfrmlsn* *subfrmlsn-size*)

(*metis* *assms(2)* *eq-iff* *in-mono* *less-imp-le* *subformulas<sub>ν</sub>-subfrmlsn* *subfrmlsn-size*)

**hence** *size-M'*:  $\bigwedge \psi. \psi \in M \implies \text{size } (\psi[N]_{\Sigma 1}) \leq \text{size } \varphi$

**and** *size-N'*:  $\bigwedge \psi. \psi \in N \implies \text{size } (\psi[M]_{\Pi 1}) \leq \text{size } \varphi$

**using** *flatten-sigma-1-length* *flatten-pi-1-length* *order-trans* **by** *blast+*

**have**  $(\sum \psi \in M. \text{size } (\psi[N]_{\Sigma 1})) \leq ?n * ?n$

**and**  $(\sum \psi \in N. \text{size } (\psi[M]_{\Pi 1})) \leq ?n * ?n$

**using** *sum-bounded-above*[*of*  $M$ , *OF* *size-M*] *sum-bounded-above*[*of*  $N$ , *OF* *size-N*]

**using** *mult-le-mono*[*OF* *card-M*] *mult-le-mono*[*OF* *card-N*] **by** *fastforce+*

**hence**  $(\sum \psi \in M. (\text{size } (\psi[N]_{\Sigma 1}) + 2)) \leq ?n * (?n + 2)$

**and**  $(\sum \psi \in N. (\text{size } (\psi[M]_{\Pi 1}) + 2)) \leq ?n * (?n + 2)$

**unfolding** *sum-associative*[*OF* *finite-M*] *sum-associative*[*OF* *finite-N*]

**using** *card-M* *card-N* **by** *simp-all*

**hence** *normal-form-disjunct-with-flatten-sigma-2-length*  $\varphi$   $M N \leq ?b$

**and** *normal-form-disjunct-with-flatten-pi-2-length*  $\varphi$   $M N \leq ?b$   
**unfolding** *normal-form-disjunct-with-flatten-sigma-2-length-def normal-form-disjunct-with-flatten-*  
*add-le-mono*)+  
**by** (*metis (no-types, lifting) flatten-sigma-2-length flatten-pi-2-length*  
*add-le-mono*)+  
**thus** *?thesis1* **and** *?thesis2*  
**by** (*simp-all add: power2-eq-square*)  
**qed**

**theorem** *normal-form-length-upper-bound:*  
*normal-form-with-flatten-sigma-2-length*  $\varphi \leq$  *normal-form-length-upper-bound*  
 $\varphi$  (**is** *?thesis1*)  
*normal-form-with-flatten-pi-2-length*  $\varphi \leq$  *normal-form-length-upper-bound*  
 $\varphi$  (**is** *?thesis2*)  
**proof** –  
**let** *?n = size*  $\varphi$   
**let** *?b = 2 ^ (size*  $\varphi + 1) + 2 * (size$   $\varphi + 2) ^ 2$

**have**  $\{(M, N) \mid M N. M \subseteq \text{subformulas}_\mu \varphi \wedge N \subseteq \text{subformulas}_\nu \varphi\} =$   
 $\{M. M \subseteq \text{subformulas}_\mu \varphi\} \times \{N. N \subseteq \text{subformulas}_\nu \varphi\}$  (**is** *?choices = -*)  
**by** *simp*

**moreover**

**have**  $\text{card } \{M. M \subseteq \text{subformulas}_\mu \varphi\} = (2 :: \text{nat}) ^ (\text{card } (\text{subformulas}_\mu \varphi))$   
**and**  $\text{card } \{N. N \subseteq \text{subformulas}_\nu \varphi\} = (2 :: \text{nat}) ^ (\text{card } (\text{subformulas}_\nu \varphi))$   
**using** *card-Pow unfolding Pow-def using subformulas $_\mu$ -finite subformulas $_\nu$ -finite* **by** *auto*

**ultimately**

**have**  $\text{card } ?\text{choices} \leq 2 ^ (\text{card } (\text{subfrmlsn } \varphi))$  (**is** *?f ≤ -*)  
**by** (*metis subformulas $_{\mu\nu}$ -card card-cartesian-product subformulas $_{\mu\nu}$ -subfrmlsn*  
*subfrmlsn-finite Suc-1 card-mono lessI power-add power-increasing-iff*)

**moreover**

**have**  $(2 :: \text{nat}) ^ (\text{card } (\text{subfrmlsn } \varphi)) \leq 2 ^ ?n$   
**using** *power-increasing[of - - 2 :: nat]* **by** (*simp add: subfrmlsn-card*)

**ultimately**

**have** *bar*: *of-nat (card ?choices) ≤ (2 :: nat) ^ ?n*  
**using** *of-nat-id* **by** *presburger*

**moreover**

**have** *normal-form-with-flatten-sigma-2-length*  $\varphi \leq \text{of-nat (card ?choices)}$   
\* *?b*  
**unfolding** *normal-form-with-flatten-sigma-2-length-def*  
**by** (*rule sum-bounded-above*) (*insert normal-form-disjunct-length-upper-bound,*  
*auto*)

**moreover**

**have** *normal-form-with-flatten-pi-2-length*  $\varphi \leq \text{of-nat (card ?choices) * ?b}$   
**unfolding** *normal-form-with-flatten-pi-2-length-def*  
**by** (*rule sum-bounded-above*) (*insert normal-form-disjunct-length-upper-bound,*  
*auto*)

**ultimately**

**show** *?thesis1* **and** *?thesis2*  
**unfolding** *normal-form-length-upper-bound-def*  
**using** *mult-le-mono1 order-trans* **by** *blast+*  
**qed**

### 3.3 Proper Subformulas

We prove that the number of (proper) subformulas (sf) in a disjunct is linear and not exponential.

**fun** *sf* :: 'a *ltn*  $\Rightarrow$  'a *ltn set*

**where**

*sf* ( $\varphi \text{ and}_n \psi$ ) = *sf*  $\varphi \cup$  *sf*  $\psi$   
| *sf* ( $\varphi \text{ or}_n \psi$ ) = *sf*  $\varphi \cup$  *sf*  $\psi$   
| *sf* ( $X_n \varphi$ ) = { $X_n \varphi$ }  $\cup$  *sf*  $\varphi$   
| *sf* ( $\varphi \text{ U}_n \psi$ ) = { $\varphi \text{ U}_n \psi$ }  $\cup$  *sf*  $\varphi \cup$  *sf*  $\psi$   
| *sf* ( $\varphi \text{ R}_n \psi$ ) = { $\varphi \text{ R}_n \psi$ }  $\cup$  *sf*  $\varphi \cup$  *sf*  $\psi$   
| *sf* ( $\varphi \text{ W}_n \psi$ ) = { $\varphi \text{ W}_n \psi$ }  $\cup$  *sf*  $\varphi \cup$  *sf*  $\psi$   
| *sf* ( $\varphi \text{ M}_n \psi$ ) = { $\varphi \text{ M}_n \psi$ }  $\cup$  *sf*  $\varphi \cup$  *sf*  $\psi$   
| *sf*  $\varphi$  = {}

**lemma** *sf-finite*:

*finite (sf  $\varphi$ )*

**by** (*induction*  $\varphi$ ) *auto*

**lemma** *sf-subset-subfrmlsn*:

$sf \varphi \subseteq subfrmlsn \varphi$

**by** (*induction*  $\varphi$ ) *auto*

**lemma** *sf-size*:

$\psi \in sf \varphi \implies size \psi \leq size \varphi$

**by** (*induction*  $\varphi$ ) *auto*

**lemma** *sf-sf-subset*:

$\psi \in sf \varphi \implies sf \psi \subseteq sf \varphi$

**by** (*induction*  $\varphi$ ) *auto*

**lemma** *subfrmlsn-sf-subset*:

$\psi \in subfrmlsn \varphi \implies sf \psi \subseteq sf \varphi$

**by** (*induction*  $\varphi$ ) *auto*

**lemma** *sf-subset-insert*:

**assumes**  $sf \varphi \subseteq insert \varphi X$

**assumes**  $\psi \in subfrmlsn \varphi$

**assumes**  $\varphi \neq \psi$

**shows**  $sf \psi \subseteq X$

**proof** –

**have**  $sf \psi \subseteq sf \varphi - \{\varphi\}$

**using** *assms(2,3)* *subfrmlsn-sf-subset* *sf-size* *subfrmlsn-size* **by** *fastforce*

**thus** *?thesis*

**using** *assms(1)* **by** *auto*

**qed**

**lemma** *flatten-pi-1-sf-subset*:

$sf (\varphi[M]_{\Pi 1}) \subseteq (\bigcup \varphi \in sf \varphi. sf (\varphi[M]_{\Pi 1}))$

**by** (*induction*  $\varphi$ ) *auto*

**lemma** *flatten-sigma-1-sf-subset*:

$sf (\varphi[M]_{\Sigma 1}) \subseteq (\bigcup \varphi \in sf \varphi. sf (\varphi[M]_{\Sigma 1}))$

**by** (*induction*  $\varphi$ ) *auto*

**lemma** *flatten-sigma-2-sf-subset*:

$sf (\varphi[M]_{\Sigma 2}) \subseteq (\bigcup \psi \in sf \varphi. sf (\psi[M]_{\Sigma 2}))$

**by** (*induction*  $\varphi$ ) *auto*

**lemma** *sf-set1*:

$sf (\varphi[M]_{\Sigma 2}) \cup sf (\varphi[M]_{\Pi 1}) \subseteq (\bigcup \psi \in (sf \varphi). (sf (\psi[M]_{\Sigma 2}) \cup sf (\psi[M]_{\Pi 1})))$

by (induction  $\varphi$ ) auto

**lemma** *ltln-not-idempotent* [simp]:

$\varphi$  and<sub>n</sub>  $\psi \neq \varphi$   $\psi$  and<sub>n</sub>  $\varphi \neq \varphi \neq \varphi$  and<sub>n</sub>  $\psi \varphi \neq \psi$  and<sub>n</sub>  $\varphi$   
 $\varphi$  or<sub>n</sub>  $\psi \neq \varphi$   $\psi$  or<sub>n</sub>  $\varphi \neq \varphi \neq \varphi$  or<sub>n</sub>  $\psi \varphi \neq \psi$  or<sub>n</sub>  $\varphi$   
 $X_n \varphi \neq \varphi \neq X_n \varphi$   
 $\varphi U_n \psi \neq \varphi \neq \varphi U_n \psi \psi U_n \varphi \neq \varphi \neq \psi U_n \varphi$   
 $\varphi R_n \psi \neq \varphi \neq \varphi R_n \psi \psi R_n \varphi \neq \varphi \neq \psi R_n \varphi$   
 $\varphi W_n \psi \neq \varphi \neq \varphi W_n \psi \psi W_n \varphi \neq \varphi \neq \psi W_n \varphi$   
 $\varphi M_n \psi \neq \varphi \neq \varphi M_n \psi \psi M_n \varphi \neq \varphi \neq \psi M_n \varphi$   
 by (induction  $\varphi$ ; force)+

**lemma** *flatten-card-sf-induct*:

assumes *finite X*

assumes  $\bigwedge x. x \in X \implies sf\ x \subseteq X$

shows  $card\ (\bigcup \varphi \in X. sf\ (\varphi[N]_{\Sigma 1})) \leq card\ X$

$\wedge card\ (\bigcup \varphi \in X. sf\ (\varphi[M]_{\Pi 1})) \leq card\ X$

$\wedge card\ (\bigcup \varphi \in X. sf\ (\varphi[M]_{\Sigma 2}) \cup sf\ (\varphi[M]_{\Pi 1})) \leq 3 * card\ X$

using *assms(2)*

**proof** (induction rule: *finite-ranking-induct*[where  $f = size$ , OF  $\langle finite\ X \rangle$ ])

case  $(\psi X)$

{

  assume  $\psi \notin X$

  hence  $\bigwedge \chi. \chi \in X \implies sf\ \chi \subseteq X$

    using  $2(2,4)$  *sf-subset-subfrmlsn subfrmlsn-size* by *fastforce*

  hence  $card\ (\bigcup \varphi \in X. sf\ (\varphi[N]_{\Sigma 1})) \leq card\ X$

    and  $card\ (\bigcup \varphi \in X. sf\ (\varphi[M]_{\Pi 1})) \leq card\ X$

    and  $card\ (\bigcup \varphi \in X. sf\ (\varphi[M]_{\Sigma 2}) \cup sf\ (\varphi[M]_{\Pi 1})) \leq 3 * card\ X$

    using  $2(3)$  by *simp+*

**moreover**

  let  $?lower1 = \bigcup \varphi \in insert\ \psi\ X. sf\ (\varphi[N]_{\Sigma 1})$

  let  $?upper1 = (\bigcup \varphi \in X. sf\ (\varphi[N]_{\Sigma 1})) \cup \{\psi[N]_{\Sigma 1}\}$

  let  $?lower2 = \bigcup \varphi \in insert\ \psi\ X. sf\ (\varphi[M]_{\Pi 1})$

  let  $?upper2 = (\bigcup \varphi \in X. sf\ (\varphi[M]_{\Pi 1})) \cup \{\psi[M]_{\Pi 1}\}$

  let  $?lower3 = \bigcup \varphi \in insert\ \psi\ X. sf\ (\varphi[M]_{\Sigma 2}) \cup sf\ (\varphi[M]_{\Pi 1})$

  let  $?upper3-cases = \{\psi[M]_{\Sigma 2}, \psi[M]_{\Pi 1}\} \cup (case\ \psi\ of\ (\varphi 1\ W_n\ \varphi 2) \Rightarrow \{G_n\ (\varphi 1[M]_{\Pi 1})\} \mid (\varphi 1\ R_n\ \varphi 2) \Rightarrow \{G_n\ (\varphi 2[M]_{\Pi 1})\} \mid - \Rightarrow \{\})$

  let  $?upper3 = (\bigcup \varphi \in X. sf\ (\varphi[M]_{\Sigma 2}) \cup sf\ (\varphi[M]_{\Pi 1})) \cup ?upper3-cases$

```

have finite-upper1: finite (?upper1)
  and finite-upper2: finite (?upper2)
  and finite-upper3: finite (?upper3)
  using 2(1) sf-finite by auto (cases  $\psi$ , auto)

have  $\bigwedge x y. \text{card } \{x, y\} \leq 3$ 
  and  $\bigwedge x y z. \text{card } \{x, y, z\} \leq 3$ 
  by (simp add: card-insert-if le-less)+
hence card-leq-3: card (?upper3-cases)  $\leq 3$ 
  by (cases  $\psi$ ) (simp-all, fast)

note card-subset-split-rule = le-trans[OF card-mono card-Un-le]

have sf-in-X: sf  $\psi \subseteq \text{insert } \psi X$ 
  using 2 by blast

have ?lower1  $\subseteq$  ?upper1  $\wedge$  ?lower2  $\subseteq$  ?upper2  $\wedge$  ?lower3  $\subseteq$  ?upper3
proof (cases  $\psi$ )
  case (And-ltln  $\psi_1 \psi_2$ )
  have *: sf  $\psi_1 \subseteq X$  sf  $\psi_2 \subseteq X$ 
    by (rule sf-subset-insert[OF sf-in-X, unfolded And-ltln]; simp)+

  have (sf ( $\psi[M]_{\Sigma 2}$ ))  $\subseteq$  ( $\bigcup \varphi \in X. \text{sf } (\varphi[M]_{\Sigma 2})$ )
  and (sf ( $\psi[M]_{\Pi 1}$ ))  $\subseteq$  ( $\bigcup \varphi \in X. \text{sf } (\varphi[M]_{\Pi 1})$ )
  and (sf ( $\psi[N]_{\Sigma 1}$ ))  $\subseteq$  ( $\bigcup \varphi \in X. \text{sf } (\varphi[N]_{\Sigma 1})$ )
  subgoal
    using flatten-sigma-2-sf-subset[of - M] * by (force simp: And-ltln)
  subgoal
    using flatten-pi-1-sf-subset[of - M] * by (force simp: And-ltln)
  subgoal
    using flatten-sigma-1-sf-subset * by (force simp: And-ltln)
  done

thus ?thesis
  by blast
next
  case (Or-ltln  $\psi_1 \psi_2$ )
  have *: sf  $\psi_1 \subseteq X$  sf  $\psi_2 \subseteq X$ 
    by (rule sf-subset-insert[OF sf-in-X, unfolded Or-ltln]; simp)+

  have (sf ( $\psi[M]_{\Sigma 2}$ ))  $\subseteq$  ( $\bigcup \varphi \in X. \text{sf } (\varphi[M]_{\Sigma 2})$ )
  and (sf ( $\psi[M]_{\Pi 1}$ ))  $\subseteq$  ( $\bigcup \varphi \in X. \text{sf } (\varphi[M]_{\Pi 1})$ )
  and (sf ( $\psi[N]_{\Sigma 1}$ ))  $\subseteq$  ( $\bigcup \varphi \in X. \text{sf } (\varphi[N]_{\Sigma 1})$ )
  subgoal

```



```

    using flatten-sigma-2-sf-subset[of - M] * by (force simp: Or-ltn)
  subgoal
    using flatten-pi-1-sf-subset[of - M] * by (force simp: Or-ltn)
  subgoal
    using flatten-sigma-1-sf-subset * by (force simp: Or-ltn)
  done

thus ?thesis
  by blast
next
case (Next-ltn  $\psi_1$ )
have *:  $sf \ \psi_1 \subseteq X$ 
  by (rule sf-subset-insert[OF sf-in-X, unfolded Next-ltn]) simp-all

have (sf ( $\psi[M]_{\Sigma 2}$ ))  $\subseteq (\bigcup \varphi \in X. sf \ (\varphi[M]_{\Sigma 2})) \cup \{\psi[M]_{\Sigma 2}\}$ 
and (sf ( $\psi[M]_{\Pi 1}$ ))  $\subseteq (\bigcup \varphi \in X. sf \ (\varphi[M]_{\Pi 1})) \cup \{\psi[M]_{\Pi 1}\}$ 
and (sf ( $\psi[N]_{\Sigma 1}$ ))  $\subseteq (\bigcup \varphi \in X. sf \ (\varphi[N]_{\Sigma 1})) \cup \{\psi[N]_{\Sigma 1}\}$ 
subgoal
  using flatten-sigma-2-sf-subset[of - M] * by (force simp: Next-ltn)
subgoal
  using flatten-pi-1-sf-subset[of - M] * by (force simp: Next-ltn)
subgoal
  using flatten-sigma-1-sf-subset * by (force simp: Next-ltn)
done

thus ?thesis
  by blast
next
case (Until-ltn  $\psi_1 \ \psi_2$ )
have *:  $sf \ \psi_1 \subseteq X \ sf \ \psi_2 \subseteq X$ 
  by (rule sf-subset-insert[OF sf-in-X, unfolded Until-ltn]; simp)+

hence (sf ( $\psi[M]_{\Sigma 2}$ ))  $\subseteq (\bigcup \varphi \in X. sf \ (\varphi[M]_{\Sigma 2})) \cup \{\psi[M]_{\Sigma 2}\}$ 
and (sf ( $\psi[M]_{\Pi 1}$ ))  $\subseteq (\bigcup \varphi \in X. sf \ (\varphi[M]_{\Pi 1})) \cup \{\psi[M]_{\Pi 1}\}$ 
and (sf ( $\psi[N]_{\Sigma 1}$ ))  $\subseteq (\bigcup \varphi \in X. sf \ (\varphi[N]_{\Sigma 1})) \cup \{\psi[N]_{\Sigma 1}\}$ 
subgoal
  using flatten-sigma-2-sf-subset[of - M] * by (force simp: Until-ltn)
subgoal
  using flatten-pi-1-sf-subset[of - M] * by (force simp: Until-ltn)
subgoal
  using flatten-sigma-1-sf-subset * by (force simp: Until-ltn)
done

thus ?thesis

```

by *blast*  
 next  
 case (*Release-ltln*  $\psi_1$   $\psi_2$ )  
 have \*:  $\text{sf } \psi_1 \subseteq X \text{ sf } \psi_2 \subseteq X$   
 by (*rule sf-subset-insert*[*OF sf-in-X*, *unfolded Release-ltln*]; *simp*)+  
  
 have ( $\text{sf } (\psi[M]_{\Sigma_2}) \subseteq (\bigcup \varphi \in X. \text{sf } (\varphi[M]_{\Sigma_2})) \cup \{\psi[M]_{\Sigma_2}, G_n$   
 $\psi_2[M]_{\Pi_1}\} \cup \text{sf } (\psi_2[M]_{\Pi_1})$ )  
 and ( $\text{sf } (\psi[M]_{\Pi_1}) \subseteq (\bigcup \varphi \in X. \text{sf } (\varphi[M]_{\Pi_1})) \cup \{\psi[M]_{\Pi_1}\}$ )  
 and ( $\text{sf } (\psi[N]_{\Sigma_1}) \subseteq (\bigcup \varphi \in X. \text{sf } (\varphi[N]_{\Sigma_1})) \cup \{\psi[N]_{\Sigma_1}\}$ )  
 subgoal  
 using *flatten-sigma-2-sf-subset*[*of - M*] \* by (*force simp: Release-ltln*)  
 subgoal  
 using *flatten-pi-1-sf-subset*[*of - M*] \* by (*force simp: Release-ltln*)  
 subgoal  
 using *flatten-sigma-1-sf-subset* \* by (*force simp: Release-ltln*)  
 done  
  
 moreover  
 have  $\text{sf } (\psi_2[M]_{\Pi_1}) \subseteq (\bigcup \varphi \in X. \text{sf } \varphi[M]_{\Sigma_2} \cup \text{sf } (\varphi[M]_{\Pi_1})) \cup \{\psi[M]_{\Pi_1}\}$   
  
 using ( $\text{sf } (\psi[M]_{\Pi_1}) \subseteq (\bigcup \varphi \in X. \text{sf } (\varphi[M]_{\Pi_1})) \cup \{\psi[M]_{\Pi_1}\}$ )  
 by (*auto simp: Release-ltln*)  
  
 ultimately  
 show ?*thesis*  
 by (*simp add: Release-ltln*) *blast*  
 next  
 case (*WeakUntil-ltln*  $\psi_1$   $\psi_2$ )  
 have \*:  $\text{sf } \psi_1 \subseteq X \text{ sf } \psi_2 \subseteq X$   
 by (*rule sf-subset-insert*[*OF sf-in-X*, *unfolded WeakUntil-ltln*]; *simp*)+  
  
 have ( $\text{sf } (\psi[M]_{\Sigma_2}) \subseteq (\bigcup \varphi \in X. \text{sf } (\varphi[M]_{\Sigma_2})) \cup \{\psi[M]_{\Sigma_2}, G_n$   
 $\psi_1[M]_{\Pi_1}\} \cup \text{sf } (\psi_1[M]_{\Pi_1})$ )  
 and ( $\text{sf } (\psi[M]_{\Pi_1}) \subseteq (\bigcup \varphi \in X. \text{sf } (\varphi[M]_{\Pi_1})) \cup \{\psi[M]_{\Pi_1}\}$ )  
 and ( $\text{sf } (\psi[N]_{\Sigma_1}) \subseteq (\bigcup \varphi \in X. \text{sf } (\varphi[N]_{\Sigma_1})) \cup \{\psi[N]_{\Sigma_1}\}$ )  
 subgoal  
 using *flatten-sigma-2-sf-subset*[*of - M*] \* by (*force simp: WeakUntil-ltln*)  
 subgoal  
 using *flatten-pi-1-sf-subset*[*of - M*] \* by (*force simp: WeakUntil-ltln*)  
 subgoal  
 using *flatten-sigma-1-sf-subset* \* by (*force simp: WeakUntil-ltln*)  
 done

**moreover**  
**have**  $sf (\psi_1[M]_{\Pi 1}) \subseteq (\bigcup \varphi \in X. sf \varphi[M]_{\Sigma 2} \cup sf (\varphi[M]_{\Pi 1})) \cup \{\psi[M]_{\Pi 1}\}$   
  
**using**  $\langle sf (\psi[M]_{\Pi 1}) \rangle \subseteq (\bigcup \varphi \in X. sf (\varphi[M]_{\Pi 1})) \cup \{\psi[M]_{\Pi 1}\}$   
**by**  $(auto simp: WeakUntil-ltln)$   
  
**ultimately**  
**show**  $?thesis$   
**by**  $(simp add: WeakUntil-ltln) blast$   
**next**  
**case**  $(StrongRelease-ltln \psi_1 \psi_2)$   
**have**  $*: sf \psi_1 \subseteq X \text{ } sf \psi_2 \subseteq X$   
**by**  $(rule sf-subset-insert[OF sf-in-X, unfolded StrongRelease-ltln]; simp)+$   
  
**hence**  $(sf (\psi[M]_{\Sigma 2})) \subseteq (\bigcup \varphi \in X. sf (\varphi[M]_{\Sigma 2})) \cup \{\psi[M]_{\Sigma 2}\}$   
**and**  $(sf (\psi[M]_{\Pi 1})) \subseteq (\bigcup \varphi \in X. sf (\varphi[M]_{\Pi 1})) \cup \{\psi[M]_{\Pi 1}\}$   
**and**  $(sf (\psi[N]_{\Sigma 1})) \subseteq (\bigcup \varphi \in X. sf (\varphi[N]_{\Sigma 1})) \cup \{\psi[N]_{\Sigma 1}\}$   
**subgoal**  
**using**  $flatten-sigma-2-sf-subset[of - M] * \text{by } (force simp: StrongRelease-ltln)$   
**subgoal**  
**using**  $flatten-pi-1-sf-subset[of - M] * \text{by } (force simp: StrongRelease-ltln)$   
**subgoal**  
**using**  $flatten-sigma-1-sf-subset * \text{by } (force simp: StrongRelease-ltln)$   
**done**  
  
**thus**  $?thesis$   
**by**  $blast$   
**qed**  $auto$   
  
**hence**  $card ?lower1 \leq card (\bigcup \varphi \in X. sf (\varphi[N]_{\Sigma 1})) + 1$   
**and**  $card ?lower2 \leq card (\bigcup \varphi \in X. sf (\varphi[M]_{\Pi 1})) + 1$   
**and**  $card ?lower3 \leq card (\bigcup \varphi \in X. sf (\varphi[M]_{\Sigma 2}) \cup sf (\varphi[M]_{\Pi 1})) +$   
 $3$   
  
**using**  $card-subset-split-rule[OF finite-upper1, of ?lower1]$   
**using**  $card-subset-split-rule[OF finite-upper2, of ?lower2]$   
**using**  $card-subset-split-rule[OF finite-upper3, of ?lower3]$   
**using**  $card-leq-3 \text{ by } simp+$   
  
**moreover**  
**have**  $card (insert \psi X) = card X + 1$   
**using**  $\langle \psi \notin X \rangle \langle finite X \rangle \text{ by } simp$   
**ultimately**  
**have**  $?case$

```

    by linarith
  }
  moreover
  have  $\psi \in X \implies ?case$ 
    using 2 by (simp add: insert-absorb)
  ultimately
  show ?case
    by meson
qed simp

```

**theorem** *flatten-card-sf*:

```

  card ( $\bigcup \psi \in sf \ \varphi. sf \ (\psi[M]_{\Sigma 1})$ )  $\leq$  card (sf  $\varphi$ ) (is ?t1)
  card ( $\bigcup \psi \in sf \ \varphi. sf \ (\psi[M]_{\Pi 1})$ )  $\leq$  card (sf  $\varphi$ ) (is ?t2)
  card ( $(sf \ (\varphi[M]_{\Sigma 2}) \cup sf \ (\varphi[M]_{\Pi 1}))$ )  $\leq$  3 * card (sf  $\varphi$ ) (is ?t3)

```

**proof** –

```

  have card ( $\bigcup \psi \in sf \ \varphi. sf \ \psi[M]_{\Sigma 2} \cup sf \ (\psi[M]_{\Pi 1})$ )  $\leq$  3 * card (sf  $\varphi$ )
    using flatten-card-sf-induct[OF sf-finite sf-sf-subset] by auto
  moreover
  have card ( $(sf \ \varphi[M]_{\Sigma 2} \cup sf \ (\varphi[M]_{\Pi 1}))$ )  $\leq$  card ( $\bigcup \psi \in sf \ \varphi. sf \ \psi[M]_{\Sigma 2} \cup$ 
sf ( $\psi[M]_{\Pi 1}$ ))
    using card-mono[OF - sf-set1] sf-finite by blast
  ultimately
  show ?t1 ?t2 ?t3
    using flatten-card-sf-induct[OF sf-finite sf-sf-subset] by auto
qed

```

**corollary** *flatten-sigma-2-card-sf*:

```

  card ( $(sf \ (\varphi[M]_{\Sigma 2}))$ )  $\leq$  3 * (card (sf  $\varphi$ ))
  by (metis sf-finite order.trans[OF - flatten-card-sf(3), of card (sf (\varphi[M]_{\Sigma 2}))),
OF card-mono] finite-UnI Un-upper1)

```

**end**

## 4 Code Export

**theory** *Normal-Form-Code-Export* **imports**

```

  LTL.Code-Equations
  LTL.Rewriting
  LTL.Disjunctive-Normal-Form
  HOL.String
  Normal-Form

```

**begin**

**fun** *flatten-pi-1-list* :: *String.literal ltn*  $\Rightarrow$  *String.literal ltn list*  $\Rightarrow$  *String.literal ltn*

**where**

*flatten-pi-1-list* ( $\psi_1 U_n \psi_2$ ) *M* = (if ( $\psi_1 U_n \psi_2$ )  $\in$  set *M* then (*flatten-pi-1-list*  $\psi_1$  *M*) *W<sub>n</sub>* (*flatten-pi-1-list*  $\psi_2$  *M*) else *false<sub>n</sub>*)  
| *flatten-pi-1-list* ( $\psi_1 W_n \psi_2$ ) *M* = (*flatten-pi-1-list*  $\psi_1$  *M*) *W<sub>n</sub>* (*flatten-pi-1-list*  $\psi_2$  *M*)  
| *flatten-pi-1-list* ( $\psi_1 M_n \psi_2$ ) *M* = (if ( $\psi_1 M_n \psi_2$ )  $\in$  set *M* then (*flatten-pi-1-list*  $\psi_1$  *M*) *R<sub>n</sub>* (*flatten-pi-1-list*  $\psi_2$  *M*) else *false<sub>n</sub>*)  
| *flatten-pi-1-list* ( $\psi_1 R_n \psi_2$ ) *M* = (*flatten-pi-1-list*  $\psi_1$  *M*) *R<sub>n</sub>* (*flatten-pi-1-list*  $\psi_2$  *M*)  
| *flatten-pi-1-list* ( $\psi_1$  *and<sub>n</sub>*  $\psi_2$ ) *M* = (*flatten-pi-1-list*  $\psi_1$  *M*) *and<sub>n</sub>* (*flatten-pi-1-list*  $\psi_2$  *M*)  
| *flatten-pi-1-list* ( $\psi_1$  *or<sub>n</sub>*  $\psi_2$ ) *M* = (*flatten-pi-1-list*  $\psi_1$  *M*) *or<sub>n</sub>* (*flatten-pi-1-list*  $\psi_2$  *M*)  
| *flatten-pi-1-list* (*X<sub>n</sub>*  $\psi$ ) *M* = *X<sub>n</sub>* (*flatten-pi-1-list*  $\psi$  *M*)  
| *flatten-pi-1-list*  $\varphi$  - =  $\varphi$

**fun** *flatten-sigma-1-list* :: *String.literal ltn*  $\Rightarrow$  *String.literal ltn list*  $\Rightarrow$  *String.literal ltn*

**where**

*flatten-sigma-1-list* ( $\psi_1 U_n \psi_2$ ) *N* = (*flatten-sigma-1-list*  $\psi_1$  *N*) *U<sub>n</sub>* (*flatten-sigma-1-list*  $\psi_2$  *N*)  
| *flatten-sigma-1-list* ( $\psi_1 W_n \psi_2$ ) *N* = (if ( $\psi_1 W_n \psi_2$ )  $\in$  set *N* then *true<sub>n</sub>* else (*flatten-sigma-1-list*  $\psi_1$  *N*) *U<sub>n</sub>* (*flatten-sigma-1-list*  $\psi_2$  *N*))  
| *flatten-sigma-1-list* ( $\psi_1 M_n \psi_2$ ) *N* = (*flatten-sigma-1-list*  $\psi_1$  *N*) *M<sub>n</sub>* (*flatten-sigma-1-list*  $\psi_2$  *N*)  
| *flatten-sigma-1-list* ( $\psi_1 R_n \psi_2$ ) *N* = (if ( $\psi_1 R_n \psi_2$ )  $\in$  set *N* then *true<sub>n</sub>* else (*flatten-sigma-1-list*  $\psi_1$  *N*) *M<sub>n</sub>* (*flatten-sigma-1-list*  $\psi_2$  *N*))  
| *flatten-sigma-1-list* ( $\psi_1$  *and<sub>n</sub>*  $\psi_2$ ) *N* = (*flatten-sigma-1-list*  $\psi_1$  *N*) *and<sub>n</sub>* (*flatten-sigma-1-list*  $\psi_2$  *N*)  
| *flatten-sigma-1-list* ( $\psi_1$  *or<sub>n</sub>*  $\psi_2$ ) *N* = (*flatten-sigma-1-list*  $\psi_1$  *N*) *or<sub>n</sub>* (*flatten-sigma-1-list*  $\psi_2$  *N*)  
| *flatten-sigma-1-list* (*X<sub>n</sub>*  $\psi$ ) *N* = *X<sub>n</sub>* (*flatten-sigma-1-list*  $\psi$  *N*)  
| *flatten-sigma-1-list*  $\varphi$  - =  $\varphi$

**fun** *flatten-sigma-2-list* :: *String.literal ltn*  $\Rightarrow$  *String.literal ltn list*  $\Rightarrow$  *String.literal ltn*

**where**

*flatten-sigma-2-list* ( $\varphi U_n \psi$ ) *M* = (*flatten-sigma-2-list*  $\varphi$  *M*) *U<sub>n</sub>* (*flatten-sigma-2-list*  $\psi$  *M*)  
| *flatten-sigma-2-list* ( $\varphi W_n \psi$ ) *M* = (*flatten-sigma-2-list*  $\varphi$  *M*) *U<sub>n</sub>* ((*flatten-sigma-2-list*  $\psi$  *M*) *or<sub>n</sub>* (*G<sub>n</sub>* (*flatten-pi-1-list*  $\varphi$  *M*)))  
| *flatten-sigma-2-list* ( $\varphi M_n \psi$ ) *M* = (*flatten-sigma-2-list*  $\varphi$  *M*) *M<sub>n</sub>* (*flatten-sigma-2-list*  $\psi$  *M*)

$\psi M$   
 $| \text{flatten-sigma-2-list } (\varphi R_n \psi) M = ((\text{flatten-sigma-2-list } \varphi M) \text{ or}_n (G_n (\text{flatten-pi-1-list } \psi M))) M_n (\text{flatten-sigma-2-list } \psi M)$   
 $| \text{flatten-sigma-2-list } (\varphi \text{ and}_n \psi) M = (\text{flatten-sigma-2-list } \varphi M) \text{ and}_n (\text{flatten-sigma-2-list } \psi M)$   
 $| \text{flatten-sigma-2-list } (\varphi \text{ or}_n \psi) M = (\text{flatten-sigma-2-list } \varphi M) \text{ or}_n (\text{flatten-sigma-2-list } \psi M)$   
 $| \text{flatten-sigma-2-list } (X_n \varphi) M = X_n (\text{flatten-sigma-2-list } \varphi M)$   
 $| \text{flatten-sigma-2-list } \varphi - = \varphi$

**lemma** *flatten-code-equations[simp]*:

$\varphi[\text{set } M]_{\Pi 1} = \text{flatten-pi-1-list } \varphi M$   
 $\varphi[\text{set } M]_{\Sigma 1} = \text{flatten-sigma-1-list } \varphi M$   
 $\varphi[\text{set } M]_{\Sigma 2} = \text{flatten-sigma-2-list } \varphi M$   
**by** (*induction*  $\varphi$ ) *auto*

**abbreviation** *and-list*  $\equiv \text{foldl And-ltln true}_n$

**abbreviation** *or-list*  $\equiv \text{foldl Or-ltln false}_n$

**definition** *normal-form-disjunct* ( $\varphi :: \text{String.literal ltln}$ )  $M N$

$\equiv (\text{flatten-sigma-2-list } \varphi M)$   
 $\text{and}_n (\text{and-list } (\text{map } (\lambda\psi. G_n (F_n (\text{flatten-sigma-1-list } \psi N)))) M)$   
 $\text{and}_n (\text{and-list } (\text{map } (\lambda\psi. F_n (G_n (\text{flatten-pi-1-list } \psi M)))) N))$

**definition** *normal-form* ( $\varphi :: \text{String.literal ltln}$ )

$\equiv \text{or-list } (\text{map } (\lambda(M, N). \text{normal-form-disjunct } \varphi M N)) (\text{advice-sets } \varphi)$

**lemma** *and-list-semantic*:  $w \models_n \text{and-list } xs \longleftrightarrow (\forall x \in \text{set } xs. w \models_n x)$

**by** (*induction*  $xs$  *rule: rev-induct*) *auto*

**lemma** *or-list-semantic*:  $w \models_n \text{or-list } xs \longleftrightarrow (\exists x \in \text{set } xs. w \models_n x)$

**by** (*induction*  $xs$  *rule: rev-induct*) *auto*

**theorem** *normal-form-correct*:

$w \models_n \varphi \longleftrightarrow w \models_n \text{normal-form } \varphi$

**proof**

**assume**  $w \models_n \varphi$

**then obtain**  $M N$  **where**  $M \subseteq \text{subformulas}_\mu \varphi$  **and**  $N \subseteq \text{subformulas}_\nu$

$\varphi$

**and**  $c1: w \models_n \varphi[M]_{\Sigma 2}$  **and**  $c2: \forall \psi \in M. w \models_n G_n (F_n \psi[N]_{\Sigma 1})$  **and**  
 $c3: \forall \psi \in N. w \models_n F_n (G_n \psi[M]_{\Pi 1})$

**using** *normal-form-with-flatten-sigma-2* **by** *metis*

**then obtain**  $ms ns$  **where**  $M = \text{set } ms$  **and**  $N = \text{set } ns$  **and**  $ms\text{-}ns\text{-in}$ :

$(ms, ns) \in \text{set } (\text{advice-sets } \varphi)$   
**by**  $(\text{meson advice-sets-subformulas})$   
**then have**  $w \models_n \text{normal-form-disjunct } \varphi \text{ ms ns}$   
**using**  $c1 \ c2 \ c3$  **by**  $(\text{simp add: and-list-semantic normal-form-disjunct-def})$   
**then show**  $w \models_n \text{normal-form } \varphi$   
**using**  $\text{normal-form-def or-list-semantic ms-ns-in}$  **by**  $\text{fastforce}$   
**next**  
**assume**  $w \models_n \text{normal-form } \varphi$   
**then obtain**  $ms \ ns$  **where**  $(ms, ns) \in \text{set } (\text{advice-sets } \varphi)$   
**and**  $w \models_n \text{normal-form-disjunct } \varphi \text{ ms ns}$   
**unfolding**  $\text{normal-form-def or-list-semantic}$  **by**  $\text{force}$   
**then have**  $\text{set } ms \subseteq \text{subformulas}_\mu \varphi$  **and**  $\text{set } ns \subseteq \text{subformulas}_\nu \varphi$   
**and**  $c1: w \models_n \varphi[\text{set } ms]_{\Sigma 2}$  **and**  $c2: \forall \psi \in \text{set } ms. w \models_n G_n (F_n \psi[\text{set } ns]_{\Sigma 1})$  **and**  $c3: \forall \psi \in \text{set } ns. w \models_n F_n (G_n \psi[\text{set } ms]_{\Pi 1})$   
**using**  $\text{advice-sets-element-subfrmlsn}$   
**by**  $(\text{auto simp: and-list-semantic normal-form-disjunct-def})$  **blast**  
**then show**  $w \models_n \varphi$   
**using**  $\text{normal-form-with-flatten-sigma-2}$  **by**  $\text{metis}$   
**qed**

**definition**  $\text{normal-form-with-simplifier } (\varphi :: \text{String.literal ltl})$   
 $\equiv \text{min-dnf } (\text{simplify Slow } (\text{normal-form } (\text{simplify Slow } \varphi)))$

**lemma**  $\text{ltl-semantic-min-dnf}$ :

$w \models_n \varphi \longleftrightarrow (\exists C \in \text{min-dnf } \varphi. \forall \psi. \psi \in C \longrightarrow w \models_n \psi)$  (**is**  $?lhs \longleftrightarrow ?rhs$ )

**proof**

**let**  $?M = \{\psi. w \models_n \psi\}$

**assume**  $?lhs$

**hence**  $?M \models_P \varphi$

**using**  $\text{ltl-models-equiv-prop-entailment}$  **by**  $\text{blast}$

**then obtain**  $M'$  **where**  $\text{fset } M' \subseteq ?M$  **and**  $M' \in \text{min-dnf } \varphi$

**using**  $\text{min-dnf-iff-prop-assignment-subset}$  **by**  $\text{blast}$

**thus**  $?rhs$

**by**  $(\text{meson in-mono mem-Collect-eq notin-fset})$

**next**

**let**  $?M = \{\psi. w \models_n \psi\}$

**assume**  $?rhs$

**then obtain**  $M'$  **where**  $\text{fset } M' \subseteq ?M$  **and**  $M' \in \text{min-dnf } \varphi$

**using**  $\text{notin-fset}$  **by**  $\text{fastforce}$

**hence**  $?M \models_P \varphi$

**using**  $\text{min-dnf-iff-prop-assignment-subset}$  **by**  $\text{blast}$

**thus**  $?lhs$

**using**  $\text{ltl-models-equiv-prop-entailment}$  **by**  $\text{blast}$

qed

**theorem**

$w \models_n \varphi \longleftrightarrow (\exists C \in (\text{normal-form-with-simplifier } \varphi). \forall \psi. \psi \models C \longrightarrow w \models_n \psi)$  (is ?lhs  $\longleftrightarrow$  ?rhs)

**unfolding** *normal-form-with-simplifier-def ltl- semantics-min-dnf* [symmetric]

**using** *normal-form-correct by simp*

In order to export the code run `isabelle build -D [PATH] -e`.

**export-code** *normal-form in SML*

**export-code** *normal-form-with-simplifier in SML*

end

## References

- [1] Ivana Černá and Radek Pelánek. Relating hierarchy of temporal properties to model checking. In Branislav Rován and Peter Vojtás, editors, *Mathematical Foundations of Computer Science 2003, 28th International Symposium, MFCS 2003, Bratislava, Slovakia, August 25-29, 2003, Proceedings*, volume 2747 of *Lecture Notes in Computer Science*, pages 318–327. Springer, 2003. doi:10.1007/978-3-540-45138-9\_26.
- [2] Edward Y. Chang, Zohar Manna, and Amir Pnueli. Characterization of temporal property classes. In Werner Kuich, editor, *Automata, Languages and Programming, 19th International Colloquium, ICALP92, Vienna, Austria, July 13-17, 1992, Proceedings*, volume 623 of *Lecture Notes in Computer Science*, pages 474–486. Springer, 1992. doi:10.1007/3-540-55719-9\_97.
- [3] Orna Lichtenstein, Amir Pnueli, and Lenore D. Zuck. The glory of the past. In Rohit Parikh, editor, *Logics of Programs, Conference, Brooklyn College, New York, NY, USA, June 17-19, 1985, Proceedings*, volume 193 of *Lecture Notes in Computer Science*, pages 196–218. Springer, 1985. doi:10.1007/3-540-15648-8\_16.
- [4] Salomon Sickert and Javier Esparza. An efficient normalisation procedure for linear temporal logic and very weak alternating automata. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2020, Saarbrücken, Germany, July 8-11, 2020*. ACM, 2020. doi:10.1145/3373718.3394743.



- [5] Salomon Sickert and Javier Esparza. An efficient normalisation procedure for linear temporal logic and very weak alternating automata. *CoRR*, abs/2005.00472, 2020. [arXiv:2005.00472](https://arxiv.org/abs/2005.00472).
- [6] Lenore D. Zuck. *Past Temporal Logic*. PhD thesis, The Weizmann Institute of Science, Israel, August 1986.