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

Salomon Sickert

December 14, 2021

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 φ_i and ψ_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

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 φ be an LTL formula and let Δ_2, Σ_1, Σ_2, and Π_1 be the classes of LTL formulas from Definition 2. Then φ is equivalent to the following formula from the class Δ_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 ψ to a formula from Σ_2, Σ_1, and Π_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 Σ_{i+1} is the least set containing Π_i and is closed under the application of conjunction, disjunction, and the* **X**, **U**, *and* **M** *operators.*
- *The class Π_{i+1} is the least set containing Σ_i and is closed under the application of conjunction, disjunction, and the* **X**, **R**, *and* **W** *operators.*
- *The class Δ_{i+1} is the least set containing Σ_{i+1} and Π_{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 *ltln-strong-weak-2*:
$w \models_n \varphi\ U_n\ \psi \longleftrightarrow w \models_n (\varphi\ and_n\ F_n\ \psi)\ W_n\ \psi$ (**is** *?thesis1*)
$w \models_n \varphi\ M_n\ \psi \longleftrightarrow w \models_n \varphi\ R_n\ (\psi\ and_n\ F_n\ \varphi)$ (**is** *?thesis2*)
proof −
have $\exists j.\ suffix\ (i + j)\ w \models_n \psi$
if $suffix\ j\ w \models_n \psi$ **and** $\forall j{\leq}i.\ \neg\ suffix\ j\ w \models_n \psi$ **for** $i\ j$
proof
from *that* **have** $j > i$
by (*cases* $j > i$) *auto*
thus $suffix\ (i + (j - i))\ w \models_n \psi$
using *that* **by** *auto*
qed
thus *?thesis1*
unfolding *ltln-strong-weak* **by** *auto*
next
have $\exists j.\ suffix\ (i + j)\ w \models_n \varphi$
if $suffix\ j\ w \models_n \varphi$ **and** $\forall j{<}i.\ \neg\ suffix\ j\ w \models_n \varphi$ **for** $i\ j$
proof
from *that* **have** $j \geq i$
by (*cases* $j \geq i$) *auto*
thus $suffix\ (i + (j - i))\ w \models_n \varphi$
using *that* **by** *auto*
qed
thus *?thesis2*
unfolding *ltln-strong-weak* **by** *auto*
qed

lemma *ltln-weak-strong-2*:
$w \models_n \varphi\ W_n\ \psi \longleftrightarrow w \models_n \varphi\ U_n\ (\psi\ or_n\ G_n\ \varphi)$ (**is** *?thesis1*)
$w \models_n \varphi\ R_n\ \psi \longleftrightarrow w \models_n (\varphi\ or_n\ G_n\ \psi)\ M_n\ \psi$ (**is** *?thesis2*)
proof −
have $suffix\ j\ w \models_n \varphi$
if $\bigwedge j.\ j < i \Longrightarrow suffix\ j\ w \models_n \varphi$ **and** $\bigwedge j.\ suffix\ (i + j)\ w \models_n \varphi$ **for** $i\ j$
using *that(1)*[*of j*] *that(2)*[*of* $j - i$] **by** (*cases* $j < i$) *simp-all*
thus *?thesis1*
unfolding *ltln-weak-strong* **unfolding** *semantics-ltln.simps suffix-suffix*
by *blast*
next
have $suffix\ j\ w \models_n \psi$
if $\bigwedge j.\ j \leq i \Longrightarrow suffix\ j\ w \models_n \psi$ **and** $\bigwedge j.\ suffix\ (i + j)\ w \models_n \psi$ **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 Σ_1-, Σ_2-, Π_1-, and Π_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
  $(\varphi\ U_n\ \psi)[M]_{\Sigma 2} = (\varphi[M]_{\Sigma 2})\ U_n\ (\psi[M]_{\Sigma 2})$
| $(\varphi\ W_n\ \psi)[M]_{\Sigma 2} = (\varphi[M]_{\Sigma 2})\ U_n\ ((\psi[M]_{\Sigma 2})\ or_n\ (G_n\ \varphi[M]_{\Pi 1}))$
| $(\varphi\ M_n\ \psi)[M]_{\Sigma 2} = (\varphi[M]_{\Sigma 2})\ M_n\ (\psi[M]_{\Sigma 2})$
| $(\varphi\ R_n\ \psi)[M]_{\Sigma 2} = ((\varphi[M]_{\Sigma 2})\ or_n\ (G_n\ \psi[M]_{\Pi 1}))\ M_n\ (\psi[M]_{\Sigma 2})$
| $(\varphi\ and_n\ \psi)[M]_{\Sigma 2} = (\varphi[M]_{\Sigma 2})\ and_n\ (\psi[M]_{\Sigma 2})$
| $(\varphi\ or_n\ \psi)[M]_{\Sigma 2} = (\varphi[M]_{\Sigma 2})\ or_n\ (\psi[M]_{\Sigma 2})$
| $(X_n\ \varphi)[M]_{\Sigma 2} = X_n\ (\varphi[M]_{\Sigma 2})$
| $\varphi[M]_{\Sigma 2} = \varphi$

fun *flatten-pi-2* :: *'a ltln* ⇒ *'a ltln set* ⇒ *'a ltln* (-[-]$_{\Pi 2}$)
where
  $(\varphi\ W_n\ \psi)[N]_{\Pi 2} = (\varphi[N]_{\Pi 2})\ W_n\ (\psi[N]_{\Pi 2})$
| $(\varphi\ U_n\ \psi)[N]_{\Pi 2} = (\varphi[N]_{\Pi 2}\ and_n\ (F_n\ \psi[N]_{\Sigma 1}))\ W_n\ (\psi[N]_{\Pi 2})$
| $(\varphi\ R_n\ \psi)[N]_{\Pi 2} = (\varphi[N]_{\Pi 2})\ R_n\ (\psi[N]_{\Pi 2})$
| $(\varphi\ M_n\ \psi)[N]_{\Pi 2} = (\varphi[N]_{\Pi 2})\ R_n\ ((\psi[N]_{\Pi 2})\ and_n\ (F_n\ \varphi[N]_{\Sigma 1}))$
| $(\varphi\ and_n\ \psi)[N]_{\Pi 2} = (\varphi[N]_{\Pi 2})\ and_n\ (\psi[N]_{\Pi 2})$
| $(\varphi\ or_n\ \psi)[N]_{\Pi 2} = (\varphi[N]_{\Pi 2})\ or_n\ (\psi[N]_{\Pi 2})$
| $(X_n\ \varphi)[N]_{\Pi 2} = X_n\ (\varphi[N]_{\Pi 2})$
| $\varphi[N]_{\Pi 2} = \varphi$

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*:
$M \cap$ *subformulas*$_\mu\ \varphi \subseteq S \Longrightarrow \varphi[M \cap S]_{\Sigma 2} = \varphi[M]_{\Sigma 2}$
by (*induction* φ) (*simp*; *blast intro*: *GF-advice-inter*)+

lemma *flatten-sigma-2-intersection-eq*:
$M \cap$ *subformulas*$_\mu\ \varphi = M' \Longrightarrow \varphi[M']_{\Sigma 2} = \varphi[M]_{\Sigma 2}$
using *flatten-sigma-2-intersection* **by** *auto*

lemma *flatten-sigma-2-monotone*:
$w \models_n \varphi[M]_{\Sigma 2} \Longrightarrow M \subseteq M' \Longrightarrow w \models_n \varphi[M']_{\Sigma 2}$
by (*induction* φ *arbitrary*: w)
(*simp*; *blast dest*: *GF-advice-monotone*)+

lemma *flatten-pi-2-intersection*:
$N \cap$ *subformulas*$_\nu\ \varphi \subseteq S \Longrightarrow \varphi[N \cap S]_{\Pi 2} = \varphi[N]_{\Pi 2}$
by (*induction* φ) (*simp*; *blast intro*: *FG-advice-inter*)+

lemma *flatten-pi-2-intersection-eq*:
$N \cap$ *subformulas*$_\nu\ \varphi = N' \Longrightarrow \varphi[N']_{\Pi 2} = \varphi[N]_{\Pi 2}$
using *flatten-pi-2-intersection* **by** *auto*

lemma *flatten-pi-2-monotone*:
$w \models_n \varphi[N]_{\Pi 2} \Longrightarrow N \subseteq N' \Longrightarrow w \models_n \varphi[N']_{\Pi 2}$
by (*induction* φ *arbitrary*: w)
(*simp*; *blast dest*: *FG-advice-monotone*)+

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

moreover

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

moreover

have $w \models_n \varphi\ U_n\ \psi \Longrightarrow$ *?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-ltln.simps*
by (*metis* $\mathcal{GF}$*-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* $\longleftrightarrow$ *?rhs*)
proof
assume *?lhs*

moreover

{

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

moreover

have $w \models_n \varphi\ M_n\ \psi \Longrightarrow$ *?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-ltln.simps*
by (*metis GF-advice-a2* $\mathcal{GF}$*-suffix order-refl*)
qed

lemma *ltln-weak-strong-stable-words*:
$w \models_n (\varphi\ W_n\ \psi) \longleftrightarrow w \models_n \varphi\ U_n\ (\psi\ or_n\ (G_n\ \varphi[\mathcal{GF}\ (\varphi\ W_n\ \psi)\ w]_{\Pi 1}))$
$w \models_n (\varphi\ R_n\ \psi) \longleftrightarrow w \models_n (\varphi\ or_n\ (G_n\ \psi[\mathcal{GF}\ (\varphi\ R_n\ \psi)\ w]_{\Pi 1}))\ M_n\ \psi$
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 $\psi \in \mathit{subfrmlsn}\ \varphi$
assumes $\mathit{suffix}\ i\ w \models_n \psi[\mathcal{GF}\ \psi\ (\mathit{suffix}\ i\ w)]_{\Sigma 2} = \mathit{suffix}\ i\ w \models_n \psi$
shows $\mathit{suffix}\ i\ w \models_n \psi[\mathcal{GF}\ \varphi\ w]_{\Sigma 2} = \mathit{suffix}\ i\ w \models_n \psi$
by (*metis* (*no-types*, *lifting*) *inf.absorb-iff2 inf-assoc inf-commute assms*(*2*) $\mathcal{GF}$*-suffix flatten-sigma-2-intersection-eq*[*of* $\mathcal{GF}\ \varphi\ w\ \psi\ \mathcal{GF}\ \psi\ w$] $\mathcal{GF}$*-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* φ *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* φ)
then show *?case*
using *flatten-sigma-2-IH-lifting*[*of* - X_n φ *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*:
$w \models_n \varphi\ U_n\ \psi \longleftrightarrow w \models_n (\varphi\ and_n\ (F_n\ \psi[\mathcal{FG}\ \psi\ w]_{\Sigma 1}))\ W_n\ \psi$ (**is** *?lhs* $\longleftrightarrow$ *?rhs*)
proof
assume *?rhs*

moreover

obtain i **where** ν-*stable* ψ (*suffix* i w)

```
    by (metis MOST-nat less-Suc-eq suffix-ν-stable)
  hence ∀ψ ∈ FG ψ w. suffix i w ⊨n Gn ψ
    using FG-suffix G-elim ν-stable-def by blast

  {
    assume assm: w ⊨n Gn (φ andn (Fn ψ[FG ψ w]Σ1))
    hence suffix i w ⊨n (Fn ψ)[FG ψ w]Σ1
      by simp
    hence suffix i w ⊨n Fn ψ
      by (blast dest: FG-advice-b2-helper[OF ‹∀ψ ∈ FG ψ w. suffix i w ⊨n
Gn ψ›])
    hence w ⊨n φ Un ψ
      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 ⋀i. suffix i w ⊨n ψ ⟹ suffix i w ⊨n ψ[FG ψ w]Σ1
    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 ⊨n φ Mn ψ ⟷ w ⊨n φ Rn (ψ andn (Fn φ[FG φ w]Σ1)) (is ?lhs ⟷
?rhs)
proof
  assume ?rhs

  moreover

  obtain i where ν-stable φ (suffix i w)
    by (metis MOST-nat less-Suc-eq suffix-ν-stable)
  hence ∀ψ ∈ FG φ w. suffix i w ⊨n Gn ψ
```

using $\mathcal{FG}$*-suffix* $\mathcal{G}$*-elim* ν*-stable-def* **by** *blast*

{
assume *assm*: $w \models_n G_n\ (\psi\ and_n\ (F_n\ \varphi[\mathcal{FG}\ \varphi\ w]_{\Sigma 1}))$
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* ‹$\forall \psi \in \mathcal{FG}\ \varphi\ w.$ *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 $\bigwedge i.$ *suffix i w* $\models_n \varphi \Longrightarrow$ *suffix i w* $\models_n \varphi[\mathcal{FG}\ \varphi\ w]_{\Sigma 1}$
using $\mathcal{FG}$*-suffix* **by** (*blast intro*: *FG-advice-b1*)

ultimately

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

lemma *ltln-strong-weak-stable-words*:
$w \models_n \varphi\ U_n\ \psi \longleftrightarrow w \models_n (\varphi\ and_n\ (F_n\ \psi[\mathcal{FG}\ (\varphi\ U_n\ \psi)\ w]_{\Sigma 1}))\ W_n\ \psi$
$w \models_n \varphi\ M_n\ \psi \longleftrightarrow w \models_n \varphi\ R_n\ (\psi\ and_n\ (F_n\ \varphi[\mathcal{FG}\ (\varphi\ M_n\ \psi)\ w]_{\Sigma 1}))$
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 $\psi \in$ *subfrmlsn* φ
assumes *suffix i w* $\models_n \psi[\mathcal{FG}\ \psi\ (\textit{suffix i w})]_{\Pi 2} =$ *suffix i w* $\models_n \psi$
shows *suffix i w* $\models_n \psi[\mathcal{FG}\ \varphi\ w]_{\Pi 2} =$ *suffix i w* $\models_n \psi$
by (*metis* (*no-types*, *lifting*) *inf.absorb-iff2 inf-assoc inf-commute assms*(*2*) $\mathcal{FG}$*-suffix flatten-pi-2-intersection-eq*[*of* $\mathcal{FG}\ \varphi\ w\ \psi\ \mathcal{FG}\ \psi\ w$] $\mathcal{FG}$*-semantics'*

subformulas$_\nu$-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* φ *arbitrary*: w)
case (*And-ltln* $\varphi 1$ $\varphi 2$)
then show *?case*
using *flatten-pi-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-pi-2-IH-lifting*[*of* - $\varphi 1$ or_n $\varphi 2$ *0*] **by** *simp*
next
case (*Next-ltln* φ)
then show *?case*
using *flatten-pi-2-IH-lifting*[*of* - X_n φ *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*:
$w \models_n \varphi \longleftrightarrow$
$(\exists\, M \subseteq \textit{subformulas}_\mu\ \varphi.\ \exists\, N \subseteq \textit{subformulas}_\nu\ \varphi.$
$w \models_n \varphi[M]_{\Sigma 2} \wedge (\forall\, \psi \in M.\ w \models_n G_n\ (F_n\ \psi[N]_{\Sigma 1})) \wedge (\forall\, \psi \in N.\ w \models_n$

$F_n\ (G_n\ \psi[M]_{\Pi 1}))$) (**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-GF* $\mathcal{FG}$*-implies-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*:
$w \models_n \varphi \longleftrightarrow$
$(\exists M \subseteq \mathit{subformulas}_\mu\ \varphi.\ \exists N \subseteq \mathit{subformulas}_\nu\ \varphi.$
$w \models_n \varphi[N]_{\Pi 2} \wedge (\forall \psi \in M.\ w \models_n G_n\ (F_n\ \psi[N]_{\Sigma 1})) \wedge (\forall \psi \in N.\ w \models_n F_n\ (G_n\ \psi[M]_{\Pi 1})))$ (**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-GF* $\mathcal{FG}$*-implies-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*:
$y > 0 \Longrightarrow y + 3 \leq (2 :: nat) \hat{} (y + 1)$
by (*induction y*) (*simp*, *fastforce*)

lemma *inequality-2*:
$x > 0 \Longrightarrow y > 0 \Longrightarrow ((2 :: nat) \hat{} (x + 1)) + (2 \hat{} (y + 1)) \leq (2 \hat{} (x + y + 1))$
by (*induction x*; *simp*; *induction y*; *simp*; *fastforce*)

lemma *size-gr-0*:
$size\ (\varphi :: 'a\ ltln) > 0$
by (*cases* φ) *simp-all*

lemma *sum-associative*:
$finite\ X \Longrightarrow (\sum x \in X.\ f\ x + c) = (\sum x \in X.\ f\ x) + card\ X * c$
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*:
$size\ (\varphi[N]_{\Sigma 1}) \leq size\ \varphi$
by (*induction* φ) *simp-all*

lemma *flatten-pi-1-length*:
$size\ (\varphi[M]_{\Pi 1}) \leq size\ \varphi$
by (*induction* φ) *simp-all*

lemma *flatten-sigma-2-length*:
$size\ (\varphi[M]_{\Sigma 2}) \leq 2 \hat{} (size\ \varphi + 1)$

proof (*induction* φ)
case (*And-ltln* $\varphi 1$ $\varphi 2$)
hence *size* $(\varphi 1\ and_n\ \varphi 2)[M]_{\Sigma 2} \leq (2\ \widehat{}\ (size\ \varphi 1 + 1)) + (2\ \widehat{}\ (size\ \varphi 2 + 1)) + 1$
by *simp*
also
have $\ldots \leq 2\ \widehat{}\ (size\ \varphi 1 + size\ \varphi 2 + 1) + 1$
using *inequality-2*[*OF size-gr-0 size-gr-0*] **by** *simp*
also
have $\ldots \leq 2\ \widehat{}\ (size\ (\varphi 1\ and_n\ \varphi 2) + 1)$
by *simp*
finally
show *?case*.
next
case (*Or-ltln* $\varphi 1$ $\varphi 2$)
hence *size* $(\varphi 1\ or_n\ \varphi 2)[M]_{\Sigma 2} \leq (2\ \widehat{}\ (size\ \varphi 1 + 1)) + (2\ \widehat{}\ (size\ \varphi 2 + 1)) + 1$
by *simp*
also
have $\ldots \leq 2\ \widehat{}\ (size\ \varphi 1 + size\ \varphi 2 + 1) + 1$
using *inequality-2*[*OF size-gr-0 size-gr-0*] **by** *simp*
also
have $\ldots \leq 2\ \widehat{}\ (size\ (\varphi 1\ or_n\ \varphi 2) + 1)$
by *simp*
finally
show *?case*.
next
case (*Next-ltln* φ)
then show *?case*
using *le-Suc-eq* **by** *fastforce*
next
case (*WeakUntil-ltln* $\varphi 1$ $\varphi 2$)
hence *size* $(\varphi 1\ W_n\ \varphi 2)[M]_{\Sigma 2} \leq 2\ \widehat{}\ (size\ \varphi 1 + 1) + 2\ \widehat{}\ (size\ \varphi 2 + 1) + size\ \varphi 1 + 4$
by (*simp*, *simp add*: *add.commute add-mono flatten-pi-1-length*)
also
have $\ldots \leq 2\ \widehat{}\ (size\ \varphi 2 + 1) + 2 * 2\ \widehat{}\ (size\ \varphi 1 + 1) + 1$
using *inequality-1*[*OF size-gr-0*, *of* $\varphi 1$] **by** *simp*
also
have $\ldots \leq 2 * (2\ \widehat{}\ (size\ \varphi 1 + 1) + 2\ \widehat{}\ (size\ \varphi 2 + 1))$
by *simp*
also
have $\ldots \leq 2 * 2\ \widehat{}\ (size\ \varphi 1 + size\ \varphi 2 + 1)$
using *inequality-2*[*OF size-gr-0 size-gr-0*] *mult-le-mono2* **by** *blast*

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

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

lemma *flatten-pi-2-length*:
size (*φ*[*N*]$_{\Pi 2}$) ≤ *2* ^ (*size φ* + *1*)
proof (*induction φ*)
case (*And-ltln φ1 φ2*)
hence *size* (*φ1* and_n *φ2*)[*N*]$_{\Pi 2}$ ≤ (*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* and_n *φ2*) + *1*)
by *simp*
finally
show *?case***.**
next
case (*Or-ltln φ1 φ2*)
hence *size* (*φ1* or_n *φ2*)[*N*]$_{\Pi 2}$ ≤ (*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* or_n *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* U_n *φ2*)[*N*]$_{\Pi 2}$ ≤ *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 U_n φ2) + 1)
    by simp
  finally
  show ?case.
next
  case (Release-ltln φ1 φ2)
  hence size (φ1 R_n 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 R_n φ2) + 1)
    by simp
  finally
  show ?case.
next
  case (WeakUntil-ltln φ1 φ2)
  hence size (φ1 W_n 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 W_n φ2) + 1)
    by simp
  finally
  show ?case.
next
  case (StrongRelease-ltln φ1 φ2)
  hence size (φ1 M_n φ2)[N]_Π2 ≤ 2 ^ (size φ1 + 1) + 2 ^ (size φ2 + 1)
```

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

definition *normal-form-length-upper-bound*
where *normal-form-length-upper-bound* φ
$\equiv (2 :: \mathit{nat}) \mathbin{\widehat{}} (\mathit{size}\ \varphi) * (2 \mathbin{\widehat{}} (\mathit{size}\ \varphi + 1) + 2 * (\mathit{size}\ \varphi + 2) \mathbin{\widehat{}} 2)$

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

definition *normal-form-with-flatten-pi-2-length*
where *normal-form-with-flatten-pi-2-length* φ
$\equiv \sum (M, N) \in \{(M, N) \mid M\ N.\ M \subseteq \mathit{subformulas}_\mu\ \varphi \wedge N \subseteq \mathit{subformulas}_\nu\ \varphi\}.$ *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* $\varphi\ M\ N$
$\equiv \mathit{size}\ (\varphi[M]_{\Sigma 2}) + (\sum \psi \in M.\ \mathit{size}\ (\psi[N]_{\Sigma 1}) + 2) + (\sum \psi \in N.\ \mathit{size}\ (\psi[M]_{\Pi 1}) + 2)$

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

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

$M \subseteq subformulas_\mu\ \varphi$
$N \subseteq subformulas_\nu\ \varphi$
shows
normal-form-disjunct-with-flatten-sigma-2-length $\varphi\ M\ N \leq 2\ \widehat{}\ (size\ \varphi + 1) + 2 * (size\ \varphi + 2)\ \widehat{}\ 2$ (**is** *?thesis1*)
normal-form-disjunct-with-flatten-pi-2-length $\varphi\ M\ N \leq 2\ \widehat{}\ (size\ \varphi + 1) + 2 * (size\ \varphi + 2)\ \widehat{}\ 2$ (**is** *?thesis2*)
proof −
let *?n* = $size\ \varphi$
let *?b* = $2\ \widehat{}\ (?n + 1) + ?n * (?n + 2) + ?n * (?n + 2)$

have *finite-M*: *finite M* **and** *card-M*: *card M* ≤ *?n*
by (*metis assms*(*1*) *finite-subset subformulas*$_\mu$*-finite*)
(*meson assms*(*1*) *card-mono order-trans subformulas*$_\mu$*-subfrmlsn subfrmlsn-card subfrmlsn-finite*)

have *finite-N*: *finite N* **and** *card-N*: *card N* ≤ *?n*
by (*metis assms*(*2*) *finite-subset subformulas*$_\nu$*-finite*)
(*meson assms*(*2*) *card-mono order-trans subformulas*$_\nu$*-subfrmlsn subfrmlsn-card subfrmlsn-finite*)

have *size-M*: $\bigwedge\psi.\ \psi \in M \Longrightarrow size\ \psi \leq size\ \varphi$
and *size-N*: $\bigwedge\psi.\ \psi \in N \Longrightarrow size\ \psi \leq size\ \varphi$
by (*metis assms*(*1*) *eq-iff in-mono less-imp-le subformulas*$_\mu$*-subfrmlsn subfrmlsn-size*)
(*metis assms*(*2*) *eq-iff in-mono less-imp-le subformulas*$_\nu$*-subfrmlsn subfrmlsn-size*)

hence *size-M′*: $\bigwedge\psi.\ \psi \in M \Longrightarrow size\ (\psi[N]_{\Sigma 1}) \leq size\ \varphi$
and *size-N′*: $\bigwedge\psi.\ \psi \in N \Longrightarrow size\ (\psi[M]_{\Pi 1}) \leq size\ \varphi$
using *flatten-sigma-1-length flatten-pi-1-length order-trans* **by** *blast+*

have $(\sum\psi \in M.\ size\ (\psi[N]_{\Sigma 1})) \leq ?n * ?n$
and $(\sum\psi \in N.\ size\ (\psi[M]_{\Pi 1})) \leq ?n * ?n$
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 $(\sum\psi \in M.\ (size\ (\psi[N]_{\Sigma 1}) + 2)) \leq ?n * (?n + 2)$
and $(\sum\psi \in N.\ (size\ (\psi[M]_{\Pi 1}) + 2)) \leq ?n * (?n + 2)$
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* φ *M N* $\leq$ *?b*
unfolding *normal-form-disjunct-with-flatten-sigma-2-length-def normal-form-disjunct-with-flatten*

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* φ (**is** *?thesis1*)
normal-form-with-flatten-pi-2-length $\varphi \leq$ *normal-form-length-upper-bound* φ (**is** *?thesis2*)
proof −
let *?n* = *size* φ
let *?b* = *2* $\hat{}$ (*size* φ + *1*) + *2* $*$ (*size* φ + *2*) $\hat{}$ *2*

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

moreover

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

ultimately

have *card ?choices* $\leq$ *2* $\hat{}$ (*card* (*subfrmlsn* φ)) (**is** *?f* $\leq$ -)
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* :: *nat*) $\hat{}$ (*card* (*subfrmlsn* φ)) $\leq$ *2* $\hat{}$ *?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* φ ≤ *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* φ ≤ *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 ltln* ⇒ *'a ltln set*
where
sf (φ and_n ψ) = *sf* φ ∪ *sf* ψ
| *sf* (φ or_n ψ) = *sf* φ ∪ *sf* ψ
| *sf* (X_n φ) = {X_n φ} ∪ *sf* φ
| *sf* (φ U_n ψ) = {φ U_n ψ} ∪ *sf* φ ∪ *sf* ψ
| *sf* (φ R_n ψ) = {φ R_n ψ} ∪ *sf* φ ∪ *sf* ψ
| *sf* (φ W_n ψ) = {φ W_n ψ} ∪ *sf* φ ∪ *sf* ψ
| *sf* (φ M_n ψ) = {φ M_n ψ} ∪ *sf* φ ∪ *sf* ψ
| *sf* φ = {}

lemma *sf-finite*:
finite (*sf* φ)

by (*induction* φ) *auto*

lemma *sf-subset-subfrmlsn*:
$sf\ \varphi \subseteq subfrmlsn\ \varphi$
by (*induction* φ) *auto*

lemma *sf-size*:
$\psi \in sf\ \varphi \Longrightarrow size\ \psi \leq size\ \varphi$
by (*induction* φ) *auto*

lemma *sf-sf-subset*:
$\psi \in sf\ \varphi \Longrightarrow sf\ \psi \subseteq sf\ \varphi$
by (*induction* φ) *auto*

lemma *subfrmlsn-sf-subset*:
$\psi \in subfrmlsn\ \varphi \Longrightarrow sf\ \psi \subseteq sf\ \varphi$
by (*induction* φ) *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* φ) *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* φ) *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* φ) *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* φ) *auto*

lemma *ltln-not-idempotent* [*simp*]:
$\varphi\ and_n\ \psi \neq \varphi\ \psi\ and_n\ \varphi \neq \varphi\ \varphi \neq \varphi\ and_n\ \psi\ \varphi \neq \psi\ and_n\ \varphi$
$\varphi\ or_n\ \psi \neq \varphi\ \psi\ or_n\ \varphi \neq \varphi\ \varphi \neq \varphi\ or_n\ \psi\ \varphi \neq \psi\ or_n\ \varphi$
$X_n\ \varphi \neq \varphi\ \varphi \neq X_n\ \varphi$
$\varphi\ U_n\ \psi \neq \varphi\ \varphi \neq \varphi\ U_n\ \psi\ \psi\ U_n\ \varphi \neq \varphi\ \varphi \neq \psi\ U_n\ \varphi$
$\varphi\ R_n\ \psi \neq \varphi\ \varphi \neq \varphi\ R_n\ \psi\ \psi\ R_n\ \varphi \neq \varphi\ \varphi \neq \psi\ R_n\ \varphi$
$\varphi\ W_n\ \psi \neq \varphi\ \varphi \neq \varphi\ W_n\ \psi\ \psi\ W_n\ \varphi \neq \varphi\ \varphi \neq \psi\ W_n\ \varphi$
$\varphi\ M_n\ \psi \neq \varphi\ \varphi \neq \varphi\ M_n\ \psi\ \psi\ M_n\ \varphi \neq \varphi\ \varphi \neq \psi\ M_n\ \varphi$
by (*induction* φ; *force*)+

lemma *flatten-card-sf-induct*:
assumes *finite X*
assumes $\bigwedge x.\ x \in X \Longrightarrow sf\ x \subseteq X$
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* ‹*finite X*›])
case (*2* ψ *X*)
{
assume $\psi \notin X$
hence $\bigwedge \chi.\ \chi \in X \Longrightarrow sf\ \chi \subseteq X$
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* = $\{\psi[M]_{\Sigma 2}, \psi[M]_{\Pi 1}\} \cup (case\ \psi\ of\ (\varphi 1\ W_n\ \varphi 2) \Rightarrow \{G_n\ (\varphi 1[M]_{\Pi 1})\} \mid (\varphi 1\ R_n\ \varphi 2) \Rightarrow \{G_n\ (\varphi 2[M]_{\Pi 1})\} \mid \text{-} \Rightarrow \{\})$
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* ψ, *auto*)

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

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

have *sf-in-X*: *sf* $\psi \subseteq$ *insert* ψ X
  using *2* **by** *blast*

have *?lower1* $\subseteq$ *?upper1* $\wedge$ *?lower2* $\subseteq$ *?upper2* $\wedge$ *?lower3* $\subseteq$ *?upper3*
proof (*cases* ψ)
  case (*And-ltln* ψ_1 ψ_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* $(\psi[M]_{\Sigma 2})) \subseteq (\bigcup \varphi \in X.$ *sf* $(\varphi[M]_{\Sigma 2}))$
    and (*sf* $(\psi[M]_{\Pi 1})) \subseteq (\bigcup \varphi \in X.$ *sf* $(\varphi[M]_{\Pi 1}))$
    and (*sf* $(\psi[N]_{\Sigma 1})) \subseteq (\bigcup \varphi \in X.$ *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* ψ_1 ψ_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* $(\psi[M]_{\Sigma 2})) \subseteq (\bigcup \varphi \in X.$ *sf* $(\varphi[M]_{\Sigma 2}))$
    and (*sf* $(\psi[M]_{\Pi 1})) \subseteq (\bigcup \varphi \in X.$ *sf* $(\varphi[M]_{\Pi 1}))$
    and (*sf* $(\psi[N]_{\Sigma 1})) \subseteq (\bigcup \varphi \in X.$ *sf* $(\varphi[N]_{\Sigma 1}))$

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

  thus ?thesis
    by blast
next
  case (Next-ltln ψ1)
  have *: sf ψ1 ⊆ X
    by (rule sf-subset-insert[OF sf-in-X, unfolded Next-ltln]) simp-all

  have (sf (ψ[M]Σ2)) ⊆ (⋃φ ∈ X. sf (φ[M]Σ2)) ∪ {ψ[M]Σ2}
    and (sf (ψ[M]Π1)) ⊆ (⋃φ ∈ X. sf (φ[M]Π1)) ∪ {ψ[M]Π1}
    and (sf (ψ[N]Σ1)) ⊆ (⋃φ ∈ X. sf (φ[N]Σ1)) ∪ {ψ[N]Σ1}
    subgoal
     using flatten-sigma-2-sf-subset[of - M] * by (force simp: Next-ltln)
    subgoal
      using flatten-pi-1-sf-subset[of - M] * by (force simp: Next-ltln)
    subgoal
      using flatten-sigma-1-sf-subset * by (force simp: Next-ltln)
    done

  thus ?thesis
    by blast
next
  case (Until-ltln ψ1 ψ2)
  have *: sf ψ1 ⊆ X sf ψ2 ⊆ X
    by (rule sf-subset-insert[OF sf-in-X, unfolded Until-ltln]; simp)+

  hence (sf (ψ[M]Σ2)) ⊆ (⋃φ ∈ X. sf (φ[M]Σ2)) ∪ {ψ[M]Σ2}
    and (sf (ψ[M]Π1)) ⊆ (⋃φ ∈ X. sf (φ[M]Π1)) ∪ {ψ[M]Π1}
    and (sf (ψ[N]Σ1)) ⊆ (⋃φ ∈ X. sf (φ[N]Σ1)) ∪ {ψ[N]Σ1}
    subgoal
    using flatten-sigma-2-sf-subset[of - M] * by (force simp: Until-ltln)
    subgoal
      using flatten-pi-1-sf-subset[of - M] * by (force simp: Until-ltln)
    subgoal
      using flatten-sigma-1-sf-subset * by (force simp: Until-ltln)
    done
```

thus *?thesis*
by *blast*
next
case (*Release-ltln* ψ_1 ψ_2)
have $*$: *sf* $\psi_1 \subseteq X$ *sf* $\psi_2 \subseteq X$
by (*rule sf-subset-insert*[*OF sf-in-X*, *unfolded Release-ltln*]; *simp*)+

have (*sf* $(\psi[M]_{\Sigma 2})) \subseteq (\bigcup \varphi \in X.\ sf\ (\varphi[M]_{\Sigma 2})) \cup \{\psi[M]_{\Sigma 2},\ G_n\ \psi_2[M]_{\Pi 1}\} \cup sf\ (\psi_2[M]_{\Pi 1})$
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*: *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 *sf* $(\psi_2[M]_{\Pi 1}) \subseteq (\bigcup \varphi \in X.\ sf\ \varphi[M]_{\Sigma 2} \cup sf\ (\varphi[M]_{\Pi 1})) \cup \{\psi[M]_{\Pi 1}\}$

using ‹(*sf* $(\psi[M]_{\Pi 1})) \subseteq (\bigcup \varphi \in X.\ sf\ (\varphi[M]_{\Pi 1})) \cup \{\psi[M]_{\Pi 1}\}$›
by (*auto simp*: *Release-ltln*)

ultimately
show *?thesis*
by (*simp add*: *Release-ltln*) *blast*
next
case (*WeakUntil-ltln* ψ_1 ψ_2)
have $*$: *sf* $\psi_1 \subseteq X$ *sf* $\psi_2 \subseteq X$
by (*rule sf-subset-insert*[*OF sf-in-X*, *unfolded WeakUntil-ltln*]; *simp*)+

have (*sf* $(\psi[M]_{\Sigma 2})) \subseteq (\bigcup \varphi \in X.\ sf\ (\varphi[M]_{\Sigma 2})) \cup \{\psi[M]_{\Sigma 2},\ G_n\ \psi_1[M]_{\Pi 1}\} \cup sf\ (\psi_1[M]_{\Pi 1})$
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*: *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 ‹$(sf\ (\psi[M]_{\Pi 1})) \subseteq (\bigcup \varphi \in X.\ sf\ (\varphi[M]_{\Pi 1})) \cup \{\psi[M]_{\Pi 1}\}$›
by (*auto simp*: *WeakUntil-ltln*)

ultimately
show *?thesis*
by (*simp add*: *WeakUntil-ltln*) *blast*
next
case (*StrongRelease-ltln* ψ_1 ψ_2)
have $*$: *sf* $\psi_1 \subseteq X$ *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*] $*$ **by** (*force simp*: *StrongRelease-ltln*)
subgoal
using *flatten-pi-1-sf-subset*[*of - M*] $*$ **by** (*force simp*: *StrongRelease-ltln*)
subgoal
using *flatten-sigma-1-sf-subset* $*$ **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* **by** *simp*+

```
    moreover
    have card (insert ψ X) = card X + 1
      using ‹ψ ∉ X› ‹finite X› by simp
    ultimately
    have ?case
      by linarith
  }
 moreover
 have ψ ∈ X ⟹ ?case
  using 2 by (simp add: insert-absorb)
 ultimately
 show ?case
  by meson
qed simp

theorem flatten-card-sf:
 card (⋃ψ ∈ sf φ. sf (ψ[M]_Σ1)) ≤ card (sf φ) (is ?t1)
 card (⋃ψ ∈ sf φ. sf (ψ[M]_Π1)) ≤ card (sf φ) (is ?t2)
 card (sf (φ[M]_Σ2) ∪ sf (φ[M]_Π1)) ≤ 3 * card (sf φ) (is ?t3)
proof −
 have card (⋃ψ ∈ sf φ. sf ψ[M]_Σ2 ∪ sf (ψ[M]_Π1)) ≤ 3 * card (sf φ)
  using flatten-card-sf-induct[OF sf-finite sf-sf-subset] by auto
 moreover
 have card (sf φ[M]_Σ2 ∪ sf (φ[M]_Π1)) ≤ card (⋃ψ ∈ sf φ. sf ψ[M]_Σ2 ∪
sf (ψ[M]_Π1))
  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 (φ[M]_Σ2)) ≤ 3 * (card (sf φ))
 by (metis sf-finite order.trans[OF - flatten-card-sf(3), of card (sf (φ[M]_Σ2)),
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 ltln* ⇒ *String.literal ltln list* ⇒ *String.literal ltln*
where
flatten-pi-1-list (ψ_1 U_n ψ_2) M = (*if* (ψ_1 U_n ψ_2) ∈ *set M then* (*flatten-pi-1-list* ψ_1 M) W_n (*flatten-pi-1-list* ψ_2 M) *else false*$_n$)
| *flatten-pi-1-list* (ψ_1 W_n ψ_2) M = (*flatten-pi-1-list* ψ_1 M) W_n (*flatten-pi-1-list* ψ_2 M)
| *flatten-pi-1-list* (ψ_1 M_n ψ_2) M = (*if* (ψ_1 M_n ψ_2) ∈ *set M then* (*flatten-pi-1-list* ψ_1 M) R_n (*flatten-pi-1-list* ψ_2 M) *else false*$_n$)
| *flatten-pi-1-list* (ψ_1 R_n ψ_2) M = (*flatten-pi-1-list* ψ_1 M) R_n (*flatten-pi-1-list* ψ_2 M)
| *flatten-pi-1-list* (ψ_1 *and*$_n$ ψ_2) M = (*flatten-pi-1-list* ψ_1 M) *and*$_n$ (*flatten-pi-1-list* ψ_2 M)
| *flatten-pi-1-list* (ψ_1 *or*$_n$ ψ_2) M = (*flatten-pi-1-list* ψ_1 M) *or*$_n$ (*flatten-pi-1-list* ψ_2 M)
| *flatten-pi-1-list* (X_n ψ) M = X_n (*flatten-pi-1-list* ψ M)
| *flatten-pi-1-list* φ - = φ

fun *flatten-sigma-1-list* :: *String.literal ltln* ⇒ *String.literal ltln list* ⇒ *String.literal ltln*
where
flatten-sigma-1-list (ψ_1 U_n ψ_2) N = (*flatten-sigma-1-list* ψ_1 N) U_n (*flatten-sigma-1-list* ψ_2 N)
| *flatten-sigma-1-list* (ψ_1 W_n ψ_2) N = (*if* (ψ_1 W_n ψ_2) ∈ *set N then true*$_n$ *else* (*flatten-sigma-1-list* ψ_1 N) U_n (*flatten-sigma-1-list* ψ_2 N))
| *flatten-sigma-1-list* (ψ_1 M_n ψ_2) N = (*flatten-sigma-1-list* ψ_1 N) M_n (*flatten-sigma-1-list* ψ_2 N)
| *flatten-sigma-1-list* (ψ_1 R_n ψ_2) N = (*if* (ψ_1 R_n ψ_2) ∈ *set N then true*$_n$ *else* (*flatten-sigma-1-list* ψ_1 N) M_n (*flatten-sigma-1-list* ψ_2 N))
| *flatten-sigma-1-list* (ψ_1 *and*$_n$ ψ_2) N = (*flatten-sigma-1-list* ψ_1 N) *and*$_n$ (*flatten-sigma-1-list* ψ_2 N)
| *flatten-sigma-1-list* (ψ_1 *or*$_n$ ψ_2) N = (*flatten-sigma-1-list* ψ_1 N) *or*$_n$ (*flatten-sigma-1-list* ψ_2 N)
| *flatten-sigma-1-list* (X_n ψ) N = X_n (*flatten-sigma-1-list* ψ N)
| *flatten-sigma-1-list* φ - = φ

fun *flatten-sigma-2-list* :: *String.literal ltln* ⇒ *String.literal ltln list* ⇒ *String.literal ltln*

where
flatten-sigma-2-list $(\varphi\ U_n\ \psi)\ M = (\mathit{flatten\text{-}sigma\text{-}2\text{-}list}\ \varphi\ M)\ U_n\ (\mathit{flatten\text{-}sigma\text{-}2\text{-}list}\ \psi\ M)$
| *flatten-sigma-2-list* $(\varphi\ W_n\ \psi)\ M = (\mathit{flatten\text{-}sigma\text{-}2\text{-}list}\ \varphi\ M)\ U_n\ ((\mathit{flatten\text{-}sigma\text{-}2\text{-}list}\ \psi\ M)\ or_n\ (G_n\ (\mathit{flatten\text{-}pi\text{-}1\text{-}list}\ \varphi\ M)))$
| *flatten-sigma-2-list* $(\varphi\ M_n\ \psi)\ M = (\mathit{flatten\text{-}sigma\text{-}2\text{-}list}\ \varphi\ M)\ M_n\ (\mathit{flatten\text{-}sigma\text{-}2\text{-}list}\ \psi\ M)$
| *flatten-sigma-2-list* $(\varphi\ R_n\ \psi)\ M = ((\mathit{flatten\text{-}sigma\text{-}2\text{-}list}\ \varphi\ M)\ or_n\ (G_n\ (\mathit{flatten\text{-}pi\text{-}1\text{-}list}\ \psi\ M)))\ M_n\ (\mathit{flatten\text{-}sigma\text{-}2\text{-}list}\ \psi\ M)$
| *flatten-sigma-2-list* $(\varphi\ and_n\ \psi)\ M = (\mathit{flatten\text{-}sigma\text{-}2\text{-}list}\ \varphi\ M)\ and_n\ (\mathit{flatten\text{-}sigma\text{-}2\text{-}list}\ \psi\ M)$
| *flatten-sigma-2-list* $(\varphi\ or_n\ \psi)\ M = (\mathit{flatten\text{-}sigma\text{-}2\text{-}list}\ \varphi\ M)\ or_n\ (\mathit{flatten\text{-}sigma\text{-}2\text{-}list}\ \psi\ M)$
| *flatten-sigma-2-list* $(X_n\ \varphi)\ M = X_n\ (\mathit{flatten\text{-}sigma\text{-}2\text{-}list}\ \varphi\ M)$
| *flatten-sigma-2-list* φ - $= \varphi$

lemma *flatten-code-equations*[*simp*]:
$\varphi[\mathit{set}\ M]_{\Pi 1} = \mathit{flatten\text{-}pi\text{-}1\text{-}list}\ \varphi\ M$
$\varphi[\mathit{set}\ M]_{\Sigma 1} = \mathit{flatten\text{-}sigma\text{-}1\text{-}list}\ \varphi\ M$
$\varphi[\mathit{set}\ M]_{\Sigma 2} = \mathit{flatten\text{-}sigma\text{-}2\text{-}list}\ \varphi\ M$
by (*induction* φ) *auto*

abbreviation *and-list* $\equiv$ *foldl And-ltln* $true_n$

abbreviation *or-list* $\equiv$ *foldl Or-ltln* $false_n$

definition *normal-form-disjunct* $(\varphi :: \mathit{String.literal\ ltln})\ M\ N$
$\equiv (\mathit{flatten\text{-}sigma\text{-}2\text{-}list}\ \varphi\ M)$
$and_n\ (\mathit{and\text{-}list}\ (\mathit{map}\ (\lambda\psi.\ G_n\ (F_n\ (\mathit{flatten\text{-}sigma\text{-}1\text{-}list}\ \psi\ N)))\ M)$
$and_n\ (\mathit{and\text{-}list}\ (\mathit{map}\ (\lambda\psi.\ F_n\ (G_n\ (\mathit{flatten\text{-}pi\text{-}1\text{-}list}\ \ \psi\ M)))\ N)))$

definition *normal-form* $(\varphi :: \mathit{String.literal\ ltln})$
$\equiv \mathit{or\text{-}list}\ (\mathit{map}\ (\lambda(M, N).\ \mathit{normal\text{-}form\text{-}disjunct}\ \varphi\ M\ N)\ (\mathit{advice\text{-}sets}\ \varphi))$

lemma *and-list-semantic*: $w \models_n \mathit{and\text{-}list}\ xs \longleftrightarrow (\forall\, x \in \mathit{set}\ xs.\ w \models_n x)$
by (*induction xs rule*: *rev-induct*) *auto*

lemma *or-list-semantic*: $w \models_n \mathit{or\text{-}list}\ xs \longleftrightarrow (\exists\, x \in \mathit{set}\ xs.\ w \models_n x)$
by (*induction xs rule*: *rev-induct*) *auto*

theorem *normal-form-correct*:
$w \models_n \varphi \longleftrightarrow w \models_n \mathit{normal\text{-}form}\ \varphi$
proof
assume $w \models_n \varphi$

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

definition *normal-form-with-simplifier* (φ :: *String.literal ltln*)
$\equiv$ *min-dnf* (*simplify Slow* (*normal-form* (*simplify Slow* φ)))

lemma *ltl-semantics-min-dnf*:
$w \models_n \varphi \longleftrightarrow (\exists C \in min\text{-}dnf\ \varphi.\ \forall \psi.\ \psi\ |\in|\ C \longrightarrow w \models_n \psi)$ (**is** *?lhs* $\longleftrightarrow$ *?rhs*)
proof
let *?M* = $\{\psi.\ w \models_n \psi\}$
assume *?lhs*
hence *?M* $\models_P \varphi$
using *ltl-models-equiv-prop-entailment* **by** *blast*
then obtain M' **where** *fset* $M' \subseteq$ *?M* **and** $M' \in min\text{-}dnf\ \varphi$
using *min-dnf-iff-prop-assignment-subset* **by** *blast*
thus *?rhs*
by (*meson in-mono mem-Collect-eq notin-fset*)
next
let *?M* = $\{\psi.\ w \models_n \psi\}$
assume *?rhs*

then obtain M' **where** *fset* $M' \subseteq$ *?M* **and** $M' \in$ *min-dnf* φ
using *notin-fset* **by** *fastforce*
hence *?M* $\models_P \varphi$
using *min-dnf-iff-prop-assignment-subset* **by** *blast*
thus *?lhs*
using *ltl-models-equiv-prop-entailment* **by** *blast*
qed

theorem
$w \models_n \varphi \longleftrightarrow (\exists\, C \in (\textit{normal-form-with-simplifier}\ \varphi).\ \forall\, \psi.\ \psi\ |\in|\ C \longrightarrow w \models_n \psi)$ (**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 Rovan 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.

[6] Lenore D. Zuck. *Past Temporal Logic*. PhD thesis, The Weizmann Institute of Science, Israel, August 1986.