Department of Mathematics, Computer Science and Physics, University of Udine

The Safety Fragment of Temporal Logics on Infinite Sequences

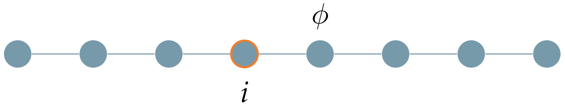
Lesson 4

Luca Geatti
luca.geatti@uniud.it
Angelo Montanari
angelo.montanari@uniud.it

April 15th, 2024

LTL+P over *finite words* is interpreted over *finite state sequences* $\sigma \in (2^{\mathcal{AP}})^+$, that is *finite, nonempty* sequences of subsets of $\mathcal{AP}$.
For the interpretation of LTL+P over finite words it suffices to consider the following cases:

- $\sigma, i \models \mathsf{X}\phi$ iff $i < |\sigma| - 1$ and $\sigma, i+1 \models \phi$

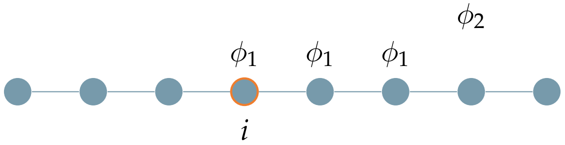


ϕ hold at the *next* position of i

Note: $\sigma, n \models \mathsf{X}\phi$ is always false when $n = |\sigma| - 1$.

LTL+P over *finite words* is interpreted over *finite state sequences* $\sigma \in (2^{\mathcal{AP}})^+$, that is *finite, nonempty* sequences of subsets of $\mathcal{AP}$.
For the interpretation of LTL+P over finite words it suffices to consider the following cases:

- $\sigma, i \models \phi_1 \mathsf{U} \phi_2$ iff $\exists i \leq j < |\sigma| \,.\, \sigma, j \models \phi_2$ and $\forall i \leq k < j \,.\, \sigma, k \models \phi_1$

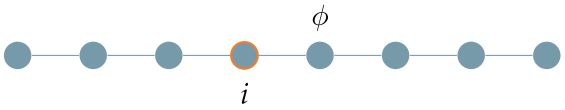


ϕ_1 holds *until* ϕ_2 holds

Shortcuts:

- (*weak tomorrow*) $\widetilde{\mathsf{X}}\phi \equiv \neg\mathsf{X}\neg\phi$

$$\sigma, i \models \widetilde{\mathsf{X}}\phi \textit{ iff } (i < |\sigma| - 1 \text{ implies } \sigma, i+1 \models \phi)$$



ϕ holds at the *next* position of i, *if any*

- **Note:** $\sigma, i \models \widetilde{\mathsf{X}}\bot$ is true iff $i = |\sigma| - 1$.
- **Note:** over *infinite traces*, X and $\widetilde{\mathsf{X}}$ coincide.

- We say that σ *satisfies* ϕ (written $\sigma \models \phi$) iff $\sigma, 0 \models \phi$.
- For any LTL+P formula ϕ, we define *the language of ϕ over finite words* as:

$$\mathcal{L}^{<\omega}(\phi) = \{\sigma \in (2^{\mathcal{AP}})^{+} \mid \sigma \models \phi\}$$

Words

- We denote with $\mathsf{LTL_f{+}P}$ the set of formulas of $\mathsf{LTL{+}P}$ that we will interpret on *finite words*

ω-Words

- We denote with $\mathsf{LTL{+}P}$ the set of formulas of $\mathsf{LTL{+}P}$ that we will interpret on *infinite words*

Words

- We denote with $\mathsf{LTL_f{+}P}$ the set of formulas of $\mathsf{LTL{+}P}$ that we will interpret on *finite words*
- We denote with $\mathsf{LTL_f}$ the set of formulas of $\mathsf{LTL_f{+}P}$ *devoid of past temporal operators*.

ω-Words

- We denote with $\mathsf{LTL{+}P}$ the set of formulas of $\mathsf{LTL{+}P}$ that we will interpret on *infinite words*
- We denote with LTL the set of formulas of $\mathsf{LTL{+}P}$ *devoid of past temporal operators*.

Words

- We denote with $\mathsf{LTL_f{+}P}$ the set of formulas of $\mathsf{LTL{+}P}$ that we will interpret on *finite words*
- We denote with $\mathsf{LTL_f}$ the set of formulas of $\mathsf{LTL_f{+}P}$ *devoid of past temporal operators*.
- Given a logic $\mathbb{L}$ (*e.g.*, $\mathsf{LTL_f}$ or $\mathsf{LTL_f{+}P}$), we denote with $[\![\mathbb{L}]\!]^{<\omega} = \{\mathcal{L}^{<\omega}(\phi) \mid \phi \in \mathbb{L}\}$

ω-Words

- We denote with $\mathsf{LTL{+}P}$ the set of formulas of $\mathsf{LTL{+}P}$ that we will interpret on *infinite words*
- We denote with LTL the set of formulas of $\mathsf{LTL{+}P}$ *devoid of past temporal operators*.
- Given a logic $\mathbb{L}$ (*e.g.*, LTL or $\mathsf{LTL{+}P}$), we denote with $[\![\mathbb{L}]\!] = \{\mathcal{L}(\phi) \mid \phi \in \mathbb{L}\}$

Theorem

- $[\![\mathsf{LTL_f{+}P}]\!]^{<\omega} = [\![\mathsf{LTL_f}]\!]^{<\omega}$
- $[\![\mathsf{LTL{+}P}]\!] = [\![\mathsf{LTL}]\!]$

Reference:

Dov M. Gabbay et al. (1980). "On the Temporal Analysis of Fairness". In: *Conference Record of the Seventh Annual ACM Symposium on Principles of Programming Languages, Las Vegas, Nevada, USA, January 1980*. Ed. by Paul W. Abrahams, Richard J. Lipton, and Stephen R. Bourne. ACM Press, pp. 163–173. URL: https://doi.org/10.1145/567446.567462

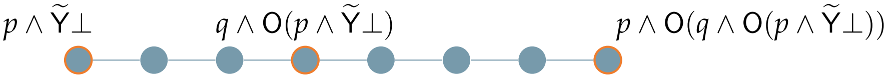
Definition (Pure-past LTL)

Pure-past LTL (pLTL, for short) is the set of LTL+P formulas *devoid* of future operators.

Example:

$p \wedge \mathsf{O}(q \wedge \mathsf{O}(p \wedge \widetilde{\mathsf{Y}}\bot))$

pLTL formulas are naturally interpreted on the *last* position of a *finite trace*.



Theorem

$[\![\mathsf{pLTL}]\!]^{<\omega} = [\![\mathsf{LTL_f}]\!]^{<\omega}$

Reference:

Orna Lichtenstein, Amir Pnueli, and Lenore Zuck (1985). "The glory of the past". In: *Workshop on Logic of Programs*. Springer, pp. 196–218. DOI: 10.1007/3-540-15648-8_16

Reference:

Lenore Zuck (1986). "Past temporal logic". In: *Weizmann Institute of Science* 67

Theorem (Kamp's Theorem over ω-words)

- *For each* LTL+P *formula* ϕ*, there exists an* S1S[FO] *formula* ψ *such that* $\mathcal{L}(\phi) = \mathcal{L}(\psi)$.
- *For each* S1S[FO] *formula* ψ*, there exists an* LTL+P *formula* ϕ *such that* $\mathcal{L}(\psi) = \mathcal{L}(\phi)$.

Theorem (Kamp's Theorem over finite words)

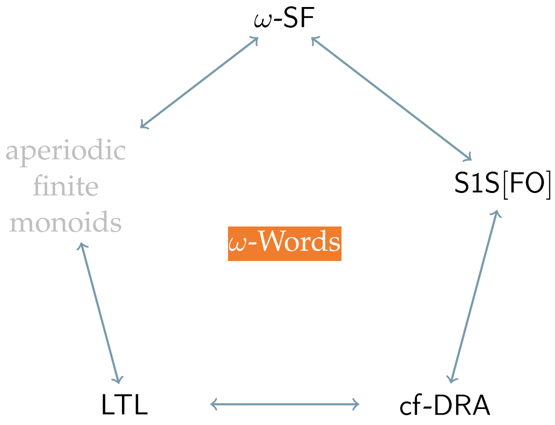
- *For each* LTL+P *formula* ϕ*, there exists an* S1S[FO] *formula* ψ *such that* $\mathcal{L}^{<\omega}(\phi) = \mathcal{L}^{<\omega}(\psi)$.
- *For each* S1S[FO] *formula* $\psi(x)$*, there exists an* LTL+P *formula* ϕ *such that* $\mathcal{L}^{<\omega}(\psi) = \mathcal{L}^{<\omega}(\phi)$.

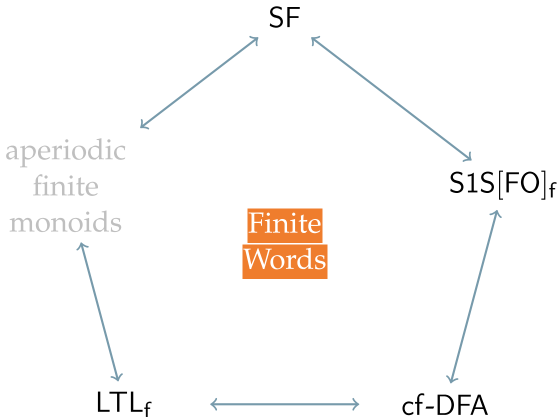
Reference:

Johan Anthony Wilem Kamp (1968). ***Tense logic and the theory of linear order.*** **University of California, Los Angeles**

Reference:

Dov M. Gabbay et al. (1980). "On the Temporal Analysis of Fairness". In: ***Conference Record of the Seventh Annual ACM Symposium on Principles of Programming Languages, Las Vegas, Nevada, USA, January 1980.*** **Ed. by Paul W. Abrahams, Richard J. Lipton, and Stephen R. Bourne. ACM Press, pp. 163–173. URL: https://doi.org/10.1145/567446.567462**


ω-SF
aperiodic
finite
monoids
S1S[FO]
ω-Words
LTL
cf-DRA


SF
aperiodic finite monoids
S1S[FO]$_f$
Finite Words
LTL$_f$
cf-DFA

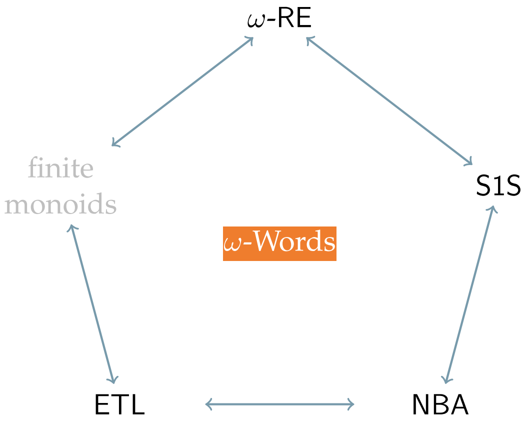
Extended Linear Temporal Logic with Past

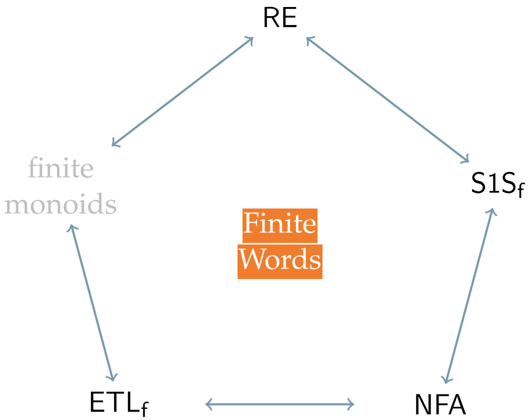
We have seen that LTL+P captures *star-free* ω-regular languages.
In order to capture all ω-regular languages, one can consider *Extended Linear Temporal Logic* (ETL, for short).

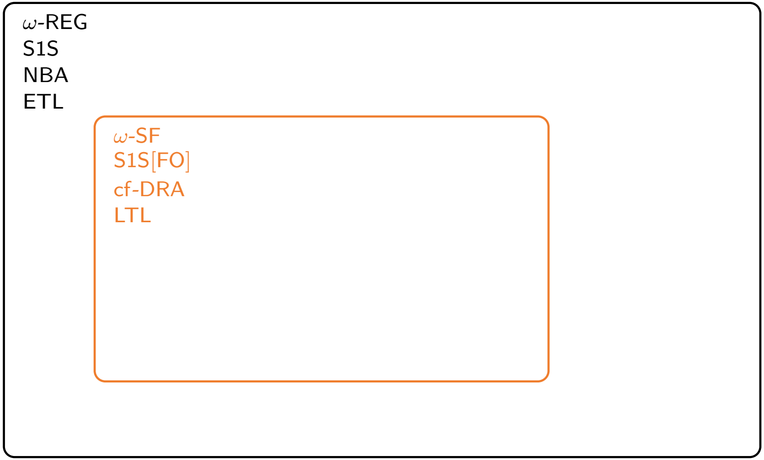
ETL = LTL + operators corresponding to *right-linear grammars*

Reference:

Pierre Wolper (1983). "Temporal logic can be more expressive". In: *Information and control* 56.1-2, pp. 72–99. DOI: 10.1016/S0019-9958(83)80051-5


ω-RE
finite monoids
S1S
ω-Words
ETL
NBA


RE
finite monoids
$S1S_f$
Finite Words
ETL_f
NFA


ω-REG
S1S
NBA
ETL
ω-SF
S1S[FO]
cf-DRA
LTL

REFERENCES

Bibliography I

Dov M. Gabbay et al. (1980). "On the Temporal Analysis of Fairness". In: *Conference Record of the Seventh Annual ACM Symposium on Principles of Programming Languages, Las Vegas, Nevada, USA, January 1980*. Ed. by Paul W. Abrahams, Richard J. Lipton, and Stephen R. Bourne. ACM Press, pp. 163–173. URL: https://doi.org/10.1145/567446.567462.

Johan Anthony Wilem Kamp (1968). ***Tense logic and the theory of linear order*.** University of California, Los Angeles.

Orna Lichtenstein, Amir Pnueli, and Lenore Zuck (1985). "The glory of the past". In: *Workshop on Logic of Programs*. Springer, pp. 196–218. DOI: 10.1007/3-540-15648-8_16.

Pierre Wolper (1983). "Temporal logic can be more expressive". In: *Information and control* 56.1-2, pp. 72–99. DOI: 10.1016/S0019-9958(83)80051-5.

Lenore Zuck (1986). "Past temporal logic". In: *Weizmann Institute of Science* 67.