Extending Freeze-LTL on Multi-Attributed Data Words with Quantifiers

Anton Pirogov

08.09.2015

Contents

Background and Related Work

- *Freeze LTL*: LTL with *freeze* and *check* operator
- Allows to store values in *registers* and compare with other values
- Checked over *data words*
- [Demri et al., 2005]: Not decidable in general!
- More than one register or *past-time operators* yield undecidability
- [Demri and Lazic, 2009]: Forward fragment with one register ($\text{LTL}_1^{\downarrow}$) decidable

A data word:

$\{p,q\}$	$\{p\}$	$\{p\}$	$\emptyset$	$\{q\}$	$\dots$
9	5	4	11	6	$\dots$

Background and Related Work

[Figueira, 2012]:

- Investigation of *Alternating Register Automata* (ARA)
- Introduction of $\exists_{\geq}$ and $\forall_{\leq}$ operators
- New, simpler proof for decidability of emptiness of ARA
- Uses proof technique based on *well-structured transition systems* [Finkel and Schnoebelen, 2001]
- $\Rightarrow$ Shows decidability of $\mathrm{LTL}_1^{\downarrow}$ with $\exists_{\geq}$ and $\forall_{\leq}$ via ARA

[Decker and Thoma, 2015]:

- Define new logic $\mathrm{LTL}_A^{\downarrow}$ based on $\mathrm{LTL}_1^{\downarrow}$
- Allows to store multiple values in each position and remains decidable!
- Requirement: *tree-quasi-ordered* attribute set and restricted comparison

Contribution of the thesis

- Take $\mathrm{LTL}_A^{\downarrow}[\mathbf{X}, \mathbf{U}]$ from [Decker and Thoma, 2015]
- Add quantifiers $\exists_{\geq}$ and $\forall_{\leq}^{x}$ from [Figueira, 2012]
- Call resulting logic $\mathrm{LTL}_A^{\downarrow}[\mathbf{X}, \mathbf{U}, \exists_{\geq}, \forall_{\leq}^{x}]$
- Adapt alternative proofs by Decker and Thoma from $\mathrm{LTL}_A^{\downarrow}[\mathbf{X}, \mathbf{U}]$ to $\mathrm{LTL}_A^{\downarrow}[\mathbf{X}, \mathbf{U}, \exists_{\geq}, \forall_{\leq}^{x}]$ to show decidability

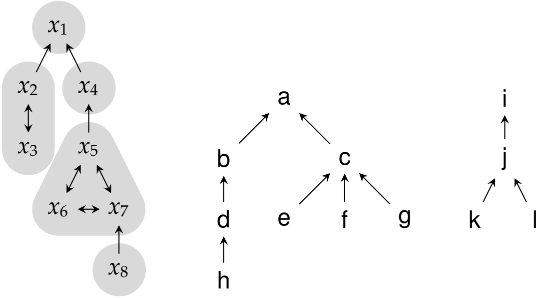
Tree-quasi-orderings

- $(A, \preccurlyeq)$ quasi-ordering + each downward-closure (*path*) total

  $\Rightarrow$ *Tree-quasi-ordering*

- All downward-closures linear $\Rightarrow$ *Tree ordering*

- min. elements = *roots*, max. elements = *leaves*



Data words

- $w = (a_1, \mathbf{d}_1)(a_2, \mathbf{d}_2) \cdots (a_n, \mathbf{d}_n) \in (\Sigma \times \Delta^A)^+$ is *A-attributed data word*
- consisting of tuples of *letters* $a_i \in \Sigma$ and *data valuations* $\mathbf{d}_i \in \Delta^A$
- map each attribute to some data value.
- If $A \approx [k]$, valuations may be called *vectors*

{a}	{b}	{a}	{c}	{b}
$x_1 \mapsto 5$	$x_1 \mapsto 5$	$x_1 \mapsto 8$	$x_1 \mapsto 8$	$x_1 \mapsto 5$
$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$
$x_2 \mapsto 3$	$x_2 \mapsto 3$	$x_2 \mapsto 3$	$x_2 \mapsto 2$	$x_2 \mapsto 2$

Syntax

Let A be fin. set of attributes, AP fin. set of atomic propositions.

Syntactically valid formulae in $\mathsf{LTL}_A^{\downarrow}[\mathbf{X}, \mathbf{U}, \exists_{\geq}, \forall^x_{\leq}]$ are described by

$$
\begin{array}{lclcccccccc}
\varphi & ::= & & p & \mid & \neg\psi & \mid & \varphi \wedge \varphi & \mid & \varphi \vee \varphi \\
 & & \mid & \mathbf{X}\varphi & \mid & \varphi \mathbf{U} \varphi & \mid & \downarrow^x \varphi & \mid & \uparrow^x \\
 & & \mid & & \exists_{\geq}\varphi & & \mid & & \forall^x_{\leq,\psi}\varphi & \\
\psi & ::= & & p & \mid & \neg\psi & \mid & \psi \wedge \psi & \mid & \psi \vee \psi \\
 & & \mid & \mathbf{X}\psi & \mid & \psi \mathbf{U} \psi & \mid & \downarrow^x \psi & \mid & \uparrow^x
\end{array}
$$

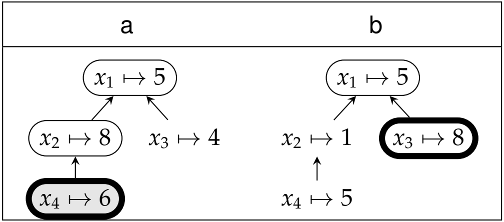
where $p \in AP$ and $x \in A$. Parentheses may be used freely.

More syntax

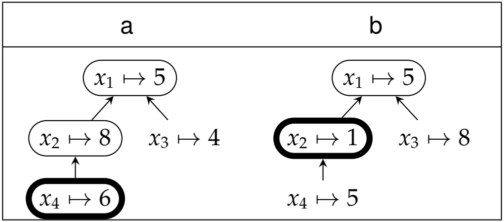
Some syntactic sugar for convenience:

$$
\begin{array}{rclrcl}
\text{true} & := & p \vee \neg p \quad p \in AP & \text{false} & := & \neg\, \text{true} \\
\varphi \Rightarrow \psi & := & \neg\varphi \vee \psi & \varphi \Leftrightarrow \psi & := & (\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi) \\
\mathbf{F}\varphi & := & \text{true}\, \mathbf{U}\varphi & \mathbf{G}\varphi & := & \neg\mathbf{F}\neg\varphi \\
\overline{\mathbf{X}}\varphi & := & \neg\mathbf{X}\neg\varphi & \varphi\, \mathbf{R}\psi & := & \neg(\neg\varphi\mathbf{U}\neg\psi)
\end{array}
$$

Example of freeze and check



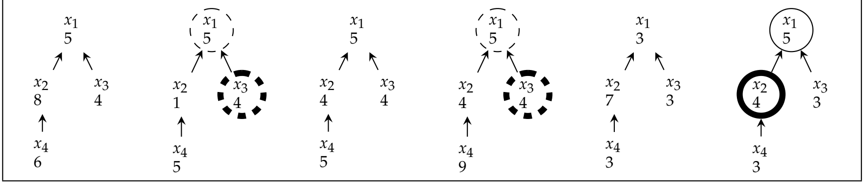
$\varphi_1 = a \wedge \downarrow^{x_4} \mathbf{X}(b \wedge \uparrow^{x_3})$



$\varphi_2 = \downarrow^{x_4} \mathbf{X} \uparrow^{x_2}$

Example of $\forall^{x}_{\leq,\psi}$ operator

1	*2*	*3*	*4*	*5*	*6*
c	a	c	a	c	b



$$\varphi = \mathbf{F}(b \wedge \forall^{x_3}_{\leq,a} \uparrow^{x_2})$$

Example of $\exists_{\geq}$ operator

b	a	b	a	a	b	a	a
1	1	3	1	3	2	2	1
↑	↑	↑	↑	↑	↑	↑	↑
2	2	2	2	2	3	3	3
↑	↑	↑	↑	↑	↑	↑	↑
3	3	1	3	1	1	1	2

$$\varphi = \exists_{\geq}((b \Rightarrow \neg \uparrow^3)\mathbf{U}(a \wedge \uparrow^3))$$

NRA execution example

$$\mathcal{A} = (\{a, b\}, 3, \{q_1, \dots, q_8\}, q_1, \delta)$$

q_i	1	2	3	4	5	6	7	8
$\delta(q_i)$	$q_2 \wedge q_3$	a	$\triangleright q_4$	$\mathsf{store}(q_5)$	$\triangleright q_6$	$q_7 \wedge q_8$	b	eq_2

$$w = \begin{array}{|ccc|} \hline \mathsf{a} & \mathsf{a} & \mathsf{b} \\ \hline 4 & 5 & 5 \\ \uparrow & \uparrow & \uparrow \\ 5 & 1 & 1 \\ \uparrow & \uparrow & \uparrow \\ 7 & 2 & 9 \\ \hline \end{array}$$

$$\Rightarrow \mathcal{C}_1 = (1, \triangleright, (a, (4, 5, 7)), \{((4, 5, 7), q_1)\})$$

$\mathcal{A}$ obviously corresponds to formula $\varphi = a \wedge \mathbf{X} \downarrow^2 \mathbf{X}(b \wedge \uparrow^2)$!

WQO and Transition Systems

Well-quasi-ordering: Let $(M, \preccurlyeq)$ be a quasi-ordering.

$(M, \preccurlyeq)$ is called *well-quasi-ordering*, if every infinite sequence of elements $m_1 m_2 m_3 \cdots$ from M contains two elements m_i, m_j, so that $m_i \preccurlyeq m_j$ and $i < j$.

Transition systems:

- $(S, \rightarrow)$ is *transition system* with set of states and trans. relation
- Succ(s) denotes the set of direct successors of state $s \in S$
- If Succ(s) is finite for all $s \in S \Rightarrow$ *finitely branching*.
- If Succ(s) is computable for all $s \in S \Rightarrow$ *effective*.

Reflexive Downward Compatibility

A transition system with a wqo relation $\leq \subset S \times S$ is called *reflexive downward compatible* wrt. $\leq$, if and only if for all $a_1, a_2, a_1' \in S$ with $a_1 \to a_2$ and $a_1' \leq a_1$ there exists a_2' with $a_2' \leq a_2$ and either $a_1' \to a_2'$ or $a_1' = a_2'$.

$$\forall a_1, a_1', a_2 \quad \exists a_2' :$$

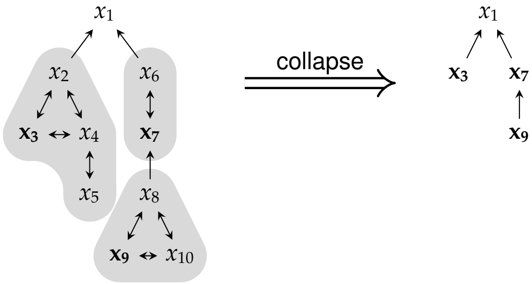
$$\begin{array}{ccccccc} a_1 & \to & a_2 & & a_1 & \to & a_2 \\ \text{|}\!\vee & & \text{|}\!\vee & \text{or} & \text{|}\!\vee & & \text{|}\!\vee \\ a_1' & \to & \mathbf{a_2'} & & a_1' & = & \mathbf{a_2'} \end{array}$$

Decidability of NRA emptiness – proof sketch

- Show that configurations of NRA give rise to a wqo:
  - Successively construct wqo relations on (parts) of configurations
  - Apply corresponding results where appropriate
- View transition relation of NRA configurations as *transition system*
  - Show that NRA transition relation is rdc wrt. the configuration wqo
  - $\Rightarrow$ effective, finitely branching *downward-WSTS with refl. compatibility*
- Apply result from [Finkel and Schnoebelen, 2001] showing that reachability of accepting configurations is decidable

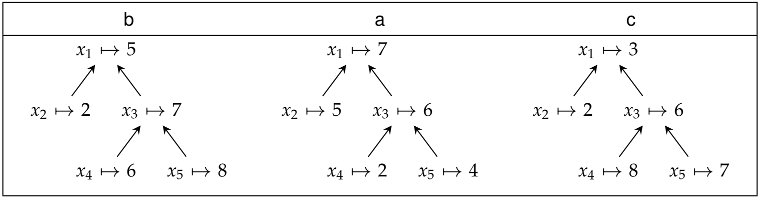
Collapsing of the SCCs

Let A be a tree-quasi-ordered set of attributes. Every LTL$^{\downarrow}_{A}$ formula φ can be translated to an equisatisfiable LTL$^{\downarrow}_{A'}$ formula φ', where A' is a tree ordering.



Frame encoding

Let A be a tree ordering, $k = \text{ht}(A)$ be the depth of A and $w \in (\Sigma \times \Delta^A)^+$ a data word of length m. We can translate the A-attributed data word w to a $[k]$-attributed data word w'.



encode ⇒

b	b	b	a	a	a	c	c	c
5	5	5	7	7	7	3	3	3
↑	↑	↑	↑	↑	↑	↑	↑	↑
2	7	7	5	6	6	2	6	6
↑	↑	↑	↑	↑	↑	↑	↑	↑
1	6	8	9	2	4	5	8	7

Linearisation - sketch

If A is a tree-quasi-ordered set of attributes of depth k, then every $\mathrm{LTL}^{\downarrow}_{A}$ formula can be translated into an equisatisfiable $\mathrm{LTL}^{\downarrow}_{[k]}$ formula.

- Let Φ be an $\mathrm{LTL}^{\downarrow}_{A}$ formula
- Applying collapsing assume formula over tree-ordered attribute set
- Argue that each model over A attributes has a model over $[k]$ attributes (frame encoding)
- Assume formula in a normal form with $\downarrow$ only before $\uparrow$ or $\mathbf{X}$ (applying some simple equivalences to "push" freeze operators to the right)
- Extend formula with additional terms to ensure that the frame encoding is correct

Linearisation - sketch

Translate formula recursively to get resulting formula $\hat{\Phi} = t(\Phi) \wedge \beta_1 \wedge \beta_2 \wedge \beta_2$:

$$
\begin{aligned}
t(a) &:= a \\
t(\neg\psi) &:= \neg t(\psi) \\
t(\psi \wedge \xi) &:= t(\psi) \wedge t(\xi) \\
t(\mathbf{X}\psi) &:= \wedge_{j=1}^{n} p_j \Rightarrow \mathbf{X}^{n-j+1} t(\psi) \\
t(\psi \mathbf{U} \xi) &:= (p_1 \Rightarrow t(\psi)) \mathbf{U} (p_1 \wedge t(\xi)) \\
t(\downarrow^x \psi) &:= \mathbf{X}^{\mathsf{sb}(x)-1} \downarrow^{\mathsf{ht}(x)} t(\psi) \\
t(\uparrow^x) &:= \wedge_{j=1}^{n} p_j \Rightarrow \mathbf{X}^{\mathsf{sb}(x)-j} \uparrow^{\mathsf{ht}(x)} \\
t(\exists_{\geq}\psi) &:= \exists_{\geq} t(\psi) \\
t(\forall^x_{\leq,\psi}\xi) &:= \forall^{\mathsf{ht}(x)}_{\leq,\psi \wedge p_{\mathsf{sb}(x)}} t(\xi)
\end{aligned}
$$

Translation to NRA - sketch

Every $\mathrm{LTL}^{\downarrow}_{[k]}[\mathbf{X}, \mathbf{U}, \exists_{\geq}, \forall^{x}_{\leq}]$ formula φ can be translated into a corresponding k-NRA $\mathcal{A}_{\varphi}$, so that for every non-empty data word $w \in (\Sigma \times \Delta^{[k]})^{+}$:

$$w \text{ satisfies } \varphi \Leftrightarrow \mathcal{A}_{\varphi} \text{ accepts } w$$

- take care of $\forall^{x}_{\leq,\psi}$: $\varphi' := \bigwedge_j \eta_j \wedge \varphi, \quad \eta_j := \mathbf{G}(\psi_j \Rightarrow \downarrow^{k} \mathbf{G} \text{ true })$
- $\Rightarrow$ carry along all data values pre-filtered by different ψ_i in formula
- convert ϕ' to negative normal form (neg. only before propostions and $\uparrow^{x}$)
- define a state for each subformula
- $\Rightarrow$ each state represents outermost token of rest-formula to verify
- define corresponding transition function

Decidability of $\mathsf{LTL}^{\downarrow}_{A}[\mathbf{X}, \mathbf{U}, \exists_{\geq}, \forall^{x}_{\leq}]$

$\mathsf{LTL}^{\downarrow}_{A}[\mathbf{X}, \mathbf{U}, \exists_{\geq}, \forall^{x}_{\leq}]$ is decidable if and only if A is a tree-quasi-ordering.

- ($\Rightarrow$) follows from [Decker and Thoma, 2015, Theorem 2]
  ($\mathsf{LTL}^{\downarrow}_{A}[\mathbf{X}, \mathbf{U}, \exists_{\geq}, \forall^{x}_{\leq}]$ is superset!)
- ($\Leftarrow$) follows from
  - linearisation
  - translation
  - decidability of NRA emptiness

Conclusion

Example formula using ordered attributes (**pid** “depends” on **res**):

$$\mathbf{G}(\mathsf{lock} \Rightarrow \downarrow^{\mathsf{pid}} ((\mathsf{use} \land \uparrow^{\mathsf{res}} \Rightarrow \uparrow^{\mathsf{pid}}) \land \neg\mathsf{halt})\mathbf{U}(\mathsf{unlock} \land \uparrow^{\mathsf{pid}}))$$

... and now we can also express properties like:

$$\exists_{\geq}\mathbf{F}((\mathsf{lock} \land \uparrow^{\mathsf{pid}}) \land \neg(\mathsf{use} \land \uparrow^{\mathsf{res}})\mathbf{U}(\mathsf{unlock} \land \uparrow^{\mathsf{pid}}))$$

“at some point a resource is locked, but not used”

$$\mathbf{G}(\mathsf{lock} \Rightarrow \forall^{\mathsf{pid}}_{\leq,\mathsf{lock}}(\uparrow^{\mathsf{res}} \Rightarrow \uparrow^{\mathsf{pid}}))$$

“each resource is always locked by the same process”

Open questions

- Lifting the logic to data trees
  - Figuiera has also results for Alternating Tree Register Automata (ATRA)
  - Add nesting to ATRA in same way?
  - Maybe interesting in context of XPath query validation
- Extending the logic with a linear ordering over the data values
  - Figuiera proved that linear ordering on data domain is decidable
  - $\Rightarrow$ Extend logic with operators like $\uparrow^x_<, \uparrow^x_>$?
  - Would allow better value inspection, e.g. $\mathbf{G}(\downarrow^{\mathsf{var}} \mathbf{X} \uparrow^{\mathsf{var}}_>)$.
- Analyzing the complexity of the logic
  - $\mathsf{LTL}^{\downarrow}_A[\mathbf{X}, \mathbf{U}]$ is not primitive-recursive
  - $\mathbf{F}_{\varepsilon_0}$-complete in *fast-growing complexity classes* [Schmitz, 2013]
  - Do $\exists_\geq, \forall^x_{\leq,\psi}$ affect this?

References I

Decker, N. and Thoma, D. (2015).
On freeze LTL with ordered attributes.
CoRR, abs/1504.06355.

Demri, S. and Lazic, R. (2009).
LTL with the freeze quantifier and register automata.
ACM Transactions on Computational Logic, 10(3).

Demri, S., Lazic, R., and Nowak, D. (2005).
On the freeze quantifier in constraint LTL: decidability and complexity.
In *12th International Symposium on Temporal Representation and Reasoning TIME 2005, 23-25 June 2005, Burlington, Vermont, USA*, pages 113–121. IEEE Computer Society.

References II

Figueira, D. (2012).

Alternating register automata on finite words and trees.

Logical Methods in Computer Science, 8(1):1–43.

Finkel, A. and Schnoebelen, P. (2001).

Well-structured transition systems everywhere!

Theoretical Computer Science, 256(1-2):63–92.

Schmitz, S. (2013).

Complexity hierarchies beyond elementary.

CoRR, abs/1312.5686.