Temporal Logics and Automata on Multi-attributed Data Words with Ordered Navigation

Normann Decker[1] Peter Habermehl[2] Martin Leucker[1]
Daniel Thoma[1]

[1]Institute for Software Engineering and Programming Languages
University of Lübeck, Germany

[2]LIAFA, CNRS, Univ Paris Diderot, Sorbonne Paris Cité, France

December 5th, 2014

Motivation

Data words appear for example in:

- Documents with data
- Behaviour with names
    - Object identifiers
    - Process identifiers

Goals:

- Specify properties
- (Runtime) Verification

(Nested) data words

Example: Lists with iterators

- Three types of identifiers
  - List identifiers: l
  - State identifiers: s
  - Iterator identifiers: i

Run of a system:

$$\begin{pmatrix} & newItr & newItr & add & newItr & next & add & \dots \\ l & 1 & 1 & 1 & 2 & 1 & 2 & \dots \\ s & 1 & 1 & 2 & 1 & 1 & 2 & \dots \\ i & 1 & 2 & 1 & 1 & 2 & 1 & \dots \end{pmatrix}$$

Properties

$$\begin{array}{c} \\ l \\ s \\ i \end{array}\left(\begin{array}{ccccccc} \mathit{newItr} & \mathit{newItr} & \mathit{add} & \mathit{newItr} & \mathit{next} & \mathit{add} & \dots \\ 1 & 1 & 1 & 2 & 1 & 2 & \dots \\ 1 & 1 & 2 & 1 & 1 & 2 & \dots \\ 1 & 2 & 1 & 1 & 2 & 1 & \dots \end{array}\right)$$

Properties:

- When an *add* occurs, the internal state of the list changes and is thus labeled by a new id: $\mathsf{G}(\mathit{add} \rightarrow \mathsf{C}_s \neg \mathsf{Y}^{=} \mathit{true})$
- When a *next* occurs, the state of the list did not change since the creation of an iterator: $\mathsf{G}(\mathit{next} \rightarrow \mathsf{C}_i(\mathit{true}\, \mathsf{S}^{=} \mathit{newItr}))$

A second example

Network printer (variation of [Bjorklund et al. 06])

- Two identifiers:
  - User: u
  - Jobs: j
- Actions: *request*, *print*, *login*, *logout*

Run of a system:

$$\begin{pmatrix} & \mathit{login} & \mathit{request} & \mathit{request} & \mathit{print} & \mathit{print} & \mathit{logout} & \mathit{login} & \dots \\ u: & 1 & 1 & 1 & 1 & 1 & 1 & 2 & \dots \\ j: & ? & 1 & 2 & 2 & 1 & ? & ? & \dots \end{pmatrix}$$

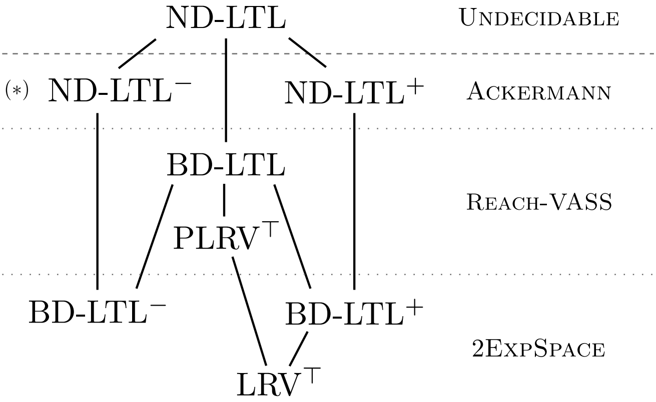
Properties:

- every print job requested by a user should eventually be printed: $\mathsf{G}(\mathit{request} \rightarrow \mathsf{C}_j\, \mathsf{X}^{=}\, \mathit{print})$
- users do not interleave: $\mathsf{G}(\mathit{login} \rightarrow (\mathsf{C}_u^1\, @u)\, \mathsf{U}\, \mathit{logout})$

Overview

- Motivation
- BD-LTL and fragments
  - Complexity of the satisfiability problem
  - Data automata and variants
- ND-LTL and fragments
  - Decidability and complexity of satisfiability problem
  - Nested data automata and variants

Overview of the logics



BD-LTL [Kara et al. 10]

Defined over multi-attributed data words, but can only bind one data value.

Syntax:

- Position formulae (LTL with past)

$$\varphi ::= p \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathsf{X}\,\varphi \mid \mathsf{Y}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi \mid \varphi\,\mathsf{S}\,\varphi \mid \mathsf{C}^{r}_{x}\,\psi$$

BD-LTL [Kara et al. 10]

Defined over multi-attributed data words, but can only bind one data value.

Syntax:

- Position formulae (LTL with past)

$$\varphi ::= p \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathsf{X}\,\varphi \mid \mathsf{Y}\,\varphi \mid \varphi\,\mathsf{U}\,\varphi \mid \varphi\,\mathsf{S}\,\varphi \mid \mathsf{C}^r_x\,\psi$$

- Class formulae

$$\psi ::= @x \mid \psi \wedge \psi \mid \neg\psi \mid \mathsf{X}^=\,\psi \mid \mathsf{Y}^=\,\psi \mid \psi\,\mathsf{U}^=\,\psi \mid \psi\,\mathsf{S}^=\,\psi \mid \varphi$$

Semantics

Formulae are interpreted over (finite) multi-attributed data words.
$\mathrm{pos}_d(w)$ is the set of positions where the data value d appears.

- $(w, i) \models \mathsf{C}^r_x\,\psi$ if $0 < i + r < |w|$ and $(w, i + r, \mathbf{d}_i(x)) \models \psi$,
- $(w, i, d) \models \varphi$ if $(w, i) \models \varphi$,
- $(w, i, d) \models @x$ if $\mathbf{d}_i(x) = d$,
- $(w, i, d) \models \mathsf{X}^{=}\,\psi$ if there is $j \in \mathrm{pos}_d(w)$, $i < j$ and, for the smallest such j, $(w, j, d) \models \psi$,
- $(w, i, d) \models \psi_1\,\mathsf{U}^{=}\,\psi_2$ if there is $j \in \mathrm{pos}_d(w)$, $i \leq j$ s. t. $(w, j, d) \models \psi_2$ and, for all $j' \in \mathrm{pos}_d(w)$, $i \leq j' < j$, $(w, j', d) \models \psi_1$.

Semantics

Formulae are interpreted over (finite) multi-attributed data words.
$\mathrm{pos}_d(w)$ is the set of positions where the data value d appears.

- $(w, i) \models \mathsf{C}^r_x\, \psi$ if $0 < i + r < |w|$ and $(w, i + r, \mathbf{d}_i(x)) \models \psi$,
- $(w, i, d) \models \varphi$ if $(w, i) \models \varphi$,
- $(w, i, d) \models @x$ if $\mathbf{d}_i(x) = d$,
- $(w, i, d) \models \mathsf{X}^{=}\, \psi$ if there is $j \in \mathrm{pos}_d(w)$, $i < j$ and, for the smallest such j, $(w, j, d) \models \psi$,
- $(w, i, d) \models \psi_1\, \mathsf{U}^{=}\, \psi_2$ if there is $j \in \mathrm{pos}_d(w)$, $i \leq j$ s. t. $(w, j, d) \models \psi_2$ and, for all $j' \in \mathrm{pos}_d(w)$, $i \leq j' < j$, $(w, j', d) \models \psi_1$.

Theorem (Kara et al. 10)

BD-LTL is decidable. As hard as reachability of VASS (Petri Nets).

Fragments

- BD-LTL^{-} (class past) : without $\mathsf{X}^{=}$ and $\mathsf{U}^{=}$
- BD-LTL^{+} (class future) : without $\mathsf{Y}^{=}$ and $\mathsf{S}^{=}$

Fragments

- BD-LTL$^-$ (class past) : without $X^=$ and $U^=$
- BD-LTL$^+$ (class future) : without $Y^=$ and $S^=$

Lemma

The satisfiability problems of both fragments are 2EXPSPACE-complete.

Proof.

Hardness: Follow from an encoding of LRV [Demri et al. 13] in BD-LTL$^+$ and from encoding of a variant in BD-LTL$^-$.
The upper bounds are shown using data automata like in [Kara et al. 10]. □

Data automata (DA)

[Bojanczyk, Segoufin, ... 2006]

- accept one-dimensional data words of $(\Sigma \times \Delta)^*$
- $\mathcal{D} = (\mathcal{A}, \mathcal{B})$
  - $\mathcal{A}$ is a letter-to-letter transducer on $\Sigma \times \Gamma$.
  - $\mathcal{B}$ is the class automaton over alphabet Γ
  - Class projection of a data word $u = \binom{a_1 a_2 \ldots a_n}{d_1 d_2 \ldots d_n}$:
    $class(d, u) = a_{i_1} a_{i_2} \ldots a_{i_k}$ s.t. $\binom{a_{i_1} a_{i_2} \ldots a_{i_k}}{d \; d \; \ldots \; d}$ is maximal subsequence of u.
    - $class(1, \binom{aab}{121}) = ab \qquad class(2, \binom{aab}{121}) = a$
    - $classes(u) := \bigcup_{d \in \Delta} class(d, u)$
  - A data word w is accepted by $\mathcal{D}$ iff $classes(\mathcal{A}(w)) \subseteq L(\mathcal{B})$.

Data automata (DA)

[Bojanczyk, Segoufin, ... 2006]

- accept one-dimensional data words of $(\Sigma \times \Delta)^*$
- $\mathcal{D} = (\mathcal{A}, \mathcal{B})$
  - $\mathcal{A}$ is a letter-to-letter transducer on $\Sigma \times \Gamma$.
  - $\mathcal{B}$ is the class automaton over alphabet Γ
  - Class projection of a data word $u = \binom{a_1 a_2 \ldots a_n}{d_1 d_2 \ldots d_n}$:
    $class(d, u) = a_{i_1} a_{i_2} \ldots a_{i_k}$ s.t. $\binom{a_{i_1} a_{i_2} \ldots a_{i_k}}{d \;\, d \;\, \ldots \;\, d}$ is maximal subsequence of u.
    - $class(1, \binom{aab}{121}) = ab$ $\quad$ $class(2, \binom{aab}{121}) = a$
    - $classes(u) := \bigcup_{d \in \Delta} class(d, u)$
  - A data word w is accepted by $\mathcal{D}$ iff $classes(\mathcal{A}(w)) \subseteq L(\mathcal{B})$.
- prefixDA: all states of $\mathcal{B}$ are final
- suffixDA: all states of $\mathcal{B}$ are initial

Emptiness of Data automata

- Translation to VASS
  - A k-dimensional VASS has transitions $q \xrightarrow{\mathbf{v}} q'$ ($\mathbf{v}$ is a k-dimensional vector over $\mathbb{Z}$)
    and $(q, \mathbf{x}) \to (q', \mathbf{x} + \mathbf{v})$ provided that $\mathbf{x} + \mathbf{v} \geq \mathbf{0}$.
  - States of the transducer $\to$ states of the VASS
  - Only the number of times each state in the class automaton B is in use has to be tracked (a counter for each state), the dimension of the VASS is the number of states of B.
- emptiness of DA: Reachability problem of VASS
- (ω)-pDA: (Repeated) control-state reachability in VASS (EXPSPACE)
- sDA: Control-state reachability in VASS (EXPSPACE) + additional stuff for ω-sDA

Satisfiability of BD-LTL and fragments

- Satisfiability of BD-LTL $\rightarrow$ Emptiness of DA (the class automaton is of exponential size)
  - tuple of data values $\rightarrow$ one data value per position (aka Umklapptrick)
  - Pure LTL $\rightarrow$ transducer of the DA
- BD-LTL^{-} $\rightarrow$ pDA of exponential size
- BD-LTL^{+} $\rightarrow$ sDA of exponential size

Tuple navigation in BD-LTL

$\mathsf{C}^r_{(x,y)}$ with $(w,i) \models \mathsf{C}^r_{(x,y)}\,\psi$ if $0 < i + r < |w|$ and $(w, i+r, (\mathbf{d}_i(x), \mathbf{d}_i(y))) \models \psi$

Theorem

The satisfiability problem of BD-LTL$^{\pm}$ with tuple navigation (it is enough to have $\mathsf{C}^r_{(x,y)}$, C^r_x and C^r_y) is undecidable.

Proof.

BD-LTL^{+} subsumes *LRV* which is known to be undecidable (PCP) when extended with tuple navigation [Demri et al. 13]. $\square$

ND-LTL: Ordered navigation

We fix an order on attributs (see examples).
Tuple navigation is restricted to data values of smaller attributes.

- $(w, i) \models \mathsf{C}_x^r\, \psi$ if $0 < i + r < |w|$ and $(w, i + r, \mathbf{d}_i|_{x<}) \models \psi$,

ND-LTL: Ordered navigation

We fix an order on attributs (see examples).
Tuple navigation is restricted to data values of smaller attributes.

- $(w, i) \models \mathsf{C}_x^r\,\psi$ if $0 < i + r < |w|$ and $(w, i + r, \mathbf{d}_i|_{x^<}) \models \psi$,
- $(w, i, \mathbf{d}) \models @x$ if $\mathbf{d}_i|_{x^<} = \mathbf{d}$,
- $(w, i, \mathbf{d}) \models \mathsf{X}^=\,\psi$ if there is $j \in \mathrm{pos}_{\mathbf{d}}(w)$, $i < j$, and, for the smallest such j, $(w, j, \mathbf{d}) \models \psi$,
- $(w, i, \mathbf{d}) \models \psi_1\,\mathsf{U}^=\,\psi_2$ if there is $j \in \mathrm{pos}_{\mathbf{d}}(w)$, $i \leq j$ s. t. $(w, j, \mathbf{d}) \models \psi_2$ and, for all $j' \in \mathrm{pos}_{\mathbf{d}}(w)$, $i \leq j' < j$, $(w, j', \mathbf{d}) \models \psi_1$.

ND-LTL: Ordered navigation

We fix an order on attributs (see examples).
Tuple navigation is restricted to data values of smaller attributes.

- $(w, i) \models \mathsf{C}_x^r\,\psi$ if $0 < i + r < |w|$ and $(w, i + r, \mathbf{d}_i|_{x^<}) \models \psi$,
- $(w, i, \mathbf{d}) \models @x$ if $\mathbf{d}_i|_{x^<} = \mathbf{d}$,
- $(w, i, \mathbf{d}) \models \mathsf{X}^=\,\psi$ if there is $j \in \mathrm{pos}_{\mathbf{d}}(w)$, $i < j$, and, for the smallest such j, $(w, j, \mathbf{d}) \models \psi$,
- $(w, i, \mathbf{d}) \models \psi_1\,\mathsf{U}^=\,\psi_2$ if there is $j \in \mathrm{pos}_{\mathbf{d}}(w)$, $i \leq j$ s. t. $(w, j, \mathbf{d}) \models \psi_2$ and, for all $j' \in \mathrm{pos}_{\mathbf{d}}(w)$, $i \leq j' < j$, $(w, j', \mathbf{d}) \models \psi_1$.

Fragments of ND-LTL

- ND-LTL$^+$, ND-LTL$^-$

Hardness results

Theorem

ND-LTL is undecidable.

Proof.

Similar to [Bjorklund, Bojanczyk 07] using two-counter machines. □

Hardness results

Theorem

ND-LTL is undecidable.

Proof.

Similar to [Bjorklund, Bojanczyk 07] using two-counter machines. □

Theorem

Satisfiability of $ND\text{-}LTL^{\pm}$ *is* ACKERMANN*-hard.*

Proof.

Using control-state reachability of lossy reset VASS. □

Hardness results

Theorem

Satisfiability of $ND\text{-}LTL^{-}$ over ω-words is undecidable.

Proof.

Using repeated control-state reachability of lossy reset VASS. $\square$

Positive results

Theorem

Satisfiability of $ND\text{-}LTL^{-}$ over finite words is decidable.

Theorem

Satisfiability of $ND\text{-}LTL^{+}$ is decidable.

Proof.

Use Nested Data Automata $\square$

Nested Data Automata

- accept *k-attribute* data words
- $\mathcal{D} = (\mathcal{A}, \mathcal{B}_1, \ldots, \mathcal{B}_k)$
  - $\mathcal{A}$ is a letter-to-letter transducer on $\Sigma \times \Gamma$
  - $\mathcal{B}_i$ are class automata over alphabet Γ
  - Now class projections are defined for each $1 \leq i \leq k$. Class projections are defined on the first i attributes.
  - A multi-attributed data word w is accepted by $\mathcal{D}$ iff for all $1 \leq i \leq k$ we have $classes_i(\mathcal{A}(w)) \subseteq L(\mathcal{B}_i)$.
- pNDA: all states of $\mathcal{B}_i$ are final
- sNDA: all states of $\mathcal{B}_i$ are initial

Example

$$\left(\begin{array}{ccccccccc} & login & req & req & print & print & logout & login & \dots \\ u: & 1 & 1 & 1 & 1 & 1 & 1 & 2 & \dots \\ j: & ? & 1 & 2 & 2 & 1 & ? & ? & \dots \end{array}\right)$$

$classes_1(w) = \{login\ req\ req\ print\ print\ logout, login \cdots, \dots\}$
$classes_2(w) = \{request\ print\}$
$L(\mathcal{B}_1) = login(req + print)^* logout$
$L(\mathcal{B}_2) = \{req\ print\}$

Results

Theorem

Emptiness of 2-NDA is undecidable.

Results

Theorem

Emptiness of 2-NDA is undecidable.

Theorem

Emptiness of pNDA and (ω)-sNDA is decidable.

Results

Theorem

Emptiness of 2-NDA is undecidable.

Theorem

Emptiness of pNDA and (ω)-sNDA is decidable.

This in turn leads to:

Theorem

Satisfiability of $ND\text{-}LTL^-$ over finite words is decidable.

Theorem

Satisfiability of $ND\text{-}LTL^+$ is decidable.

Handling NDA

NDA $\rightarrow$ nested VASS

Nested VASS (here of order 2)

- states: $(q_1, \{(q_1', \{q_1'' : 2, q_2'' : 3\}) : 2, (q_2', \{q_2'' : 3\}) : 3\})$
- transitions are of the form
  - $q_1 \rightarrow (q_2, q_1', q_1'')$
    ex: $(q_2, \{(q_1', \{q_1'' : 2, q_2'' : 3\}) : 2, (q_2', \{q_2'' : 3\}) : 3,$
    $(q_1', \{q_1'' : 1\}) : 1\})$
  - $(q_1, q_1') \rightarrow (q_2, q_2', q_2'')$
    ex: $(q_2, \{(q_1', \{q_1'' : 2, q_2'' : 3\}) : 1,\ (q_2', \{q_1'' : 2, q_2'' : 4\}) : 1),$
    $(q_2', \{q_2'' : 3\}) : 3\})$
  - $(q_1, q_1', q_1'') \rightarrow (q_2, q_2', q_2'')$
    ex: $(q_2, \{(q_1', \{q_1'' : 2, q_2'' : 3\}) : 1, (q_2', \{q_1'' : 2, q_2'' : 2\}) : 1,$
    $(q_2', \{q_2'' : 1\}) : 1, (q_2', \{q_2'' : 3\}) : 3\})$
- a set of initial control states and a set of final states (control state + others)

Results on nested VASS

Lemma

Reachability is undecidable for order 2 nested VASS.

Proof.

Simulate a 2-counter machine (similarly as [Bjorklund, Bojanczyk 07]). □

Results on nested VASS

Lemma

Reachability is undecidable for order 2 nested VASS.

Proof.

Simulate a 2-counter machine (similarly as [Bjorklund, Bojanczyk 07]). □

Lemma

Coverability is decidable.

Proof.

nested VASS are well-structured transition systems. □

From NDA to nested VASS

Example from 2-NDA to 2-nested VASS:
while reading a letter (a, d_1, d_2):

- Transducer move $(a/A) \rightarrow$ control state move
- guess:
  - d_1 is new: we guess two initial states of $\mathcal{B}_1$ and $\mathcal{B}_2$ which can perform A
  - d_1 is not new but d_2 is: we take a state in $\mathcal{B}_1$ already remembered and guess an initial state of $\mathcal{B}_2$ and match A
  - (d_1, d_2) is not new: We match the A move in both $\mathcal{B}_1$ and $\mathcal{B}_2$

Conclusion and open problems

- Decidable logics on multi-attributed data words with restricted tuple navigation
- Other decidable data logics with tuple navigation ?
- Applications