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

Normann Decker\textsuperscript{1}        Peter Habermehl\textsuperscript{2}        Martin Leucker\textsuperscript{1}        Daniel Thoma\textsuperscript{1}

\textsuperscript{1}Institute for Software Engineering and Programming Languages
University of Lübeck, Germany

\textsuperscript{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:

\[
\begin{pmatrix}
\text{newlitr} & \text{newlitr} & \text{add} & \text{newlitr} & \text{next} & \text{add} & \ldots \\
1 & 1 & 1 & 1 & 2 & 1 & 2 & \ldots \\
1 & 1 & 2 & 1 & 1 & 2 & \ldots \\
1 & 2 & 1 & 1 & 2 & 1 & \ldots \\
\end{pmatrix}
\]
Properties:

- When an `add` occurs, the internal state of the list changes and is thus labeled by a new id: $G(add \rightarrow C_s \neg Y \equiv true)$
- When a `next` occurs, the state of the list did not change since the creation of an iterator: $G(next \rightarrow C_i(true S \equiv newltr))$
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:

\[
\begin{pmatrix}
  \text{login} & \text{request} & \text{request} & \text{print} & \text{print} & \text{logout} & \text{login} & \ldots \\
  u : & 1 & 1 & 1 & 1 & 1 & 1 & 2 & \ldots \\
\end{pmatrix}
\]

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

ND-LTL

(*) ND-LTL^-

BD-LTL

PLRV^T

BD-LTL^-

BD-LTL^+

LRV^T

UNDECIDABLE

ACKERMANN

REACH-VASS

2ExpSpace
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)

\[ \varphi ::= p \mid \varphi \land \varphi \mid \neg \varphi \mid X \varphi \mid Y \varphi \mid \varphi U \varphi \mid \varphi S \varphi \mid C^r_x \psi \]
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)

\[ \varphi ::= p \mid \varphi \land \varphi \mid \neg \varphi \mid X \varphi \mid Y \varphi \mid \varphi U \varphi \mid \varphi S \varphi \mid C^r_x \psi \]

- Class formulae

\[ \psi ::= @x \mid \psi \land \psi \mid \neg \psi \mid X^= \psi \mid Y^= \psi \mid \psi U^= \psi \mid \psi S^= \psi \mid \varphi \]
Semantics

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

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

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

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

Theorem (Kara et al. 10)

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

- BD-LTL\(^{-}\) (class past) : without \(X^=\) and \(U^=\)
- BD-LTL\(^{+}\) (class future) : without \(Y^=\) and \(S^=\)
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 \(\Gamma\).
- Class projection of a data word \(u = (a_1a_2...a_n)\):
  \[\text{class}(d, u) = a_{i_1}a_{i_2}...a_{i_k} \text{ s.t. } (a_{i_1}a_{i_2}...a_{i_k}) \text{ is maximal subsequence of } u.\]
  - \(\text{class}(1, (\begin{array}{c}a\ b\
1 \\
2 \\
1 \end{array})) = ab\)
  - \(\text{class}(2, (\begin{array}{c}a\ b\
1 \\
2 \\
1 \end{array})) = a\)
- \(\text{classes}(u) := \bigcup_{d \in \Delta} \text{class}(d, u)\)
- A data word \(w\) is accepted by \(\mathcal{D}\) iff \(\text{classes}(\mathcal{A}(w)) \subseteq L(\mathcal{B})\).
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 \(\Gamma\)
  - Class projection of a data word \(u = (a_1 a_2 \ldots a_n; d_1 d_2 \ldots d_n)\):
    
    \[
    \text{class}(d, u) = a_{i_1} a_{i_2} \ldots a_{i_k} \text{ s.t. } (a_{i_1} a_{i_2} \ldots a_{i_k}; d d \ldots d) \text{ is maximal subsequence of } u.
    \]
    - \(\text{class}(1, (\frac{aab}{121})) = ab\)
    - \(\text{class}(2, (\frac{aab}{121})) = a\)
  - \(\text{classes}(u) := \bigcup_{d \in \Delta} \text{class}(d, u)\)
- A data word \(w\) is accepted by \(\mathcal{D}\) iff \(\text{classes}(\mathcal{A}(w)) \subseteq L(\mathcal{B})\).
  
  - **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 $(q, x) \rightarrow (q', x + \mathbf{v})$ provided that $x + \mathbf{v} \geq \mathbf{0}$.
  - States of the transducer $\rightarrow$ 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
- $(\omega)$-pDA: (Repeated) control-state reachability in VASS (EXPSPACE)
- sDA: Control-state reachability in VASS (EXPSPACE) + additional stuff for $\omega$-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

\[ C^r_{(x,y)} (w, i) \models C^r_{(x,y)} \psi \text{ if } 0 < i + r < |w| \]
\[ \text{and } (w, i + r, (d_i(x), d_i(y))) \models \psi \]

**Theorem**

The satisfiability problem of \( BD-LTL^\pm \) with tuple navigation (it is enough to have \( C^r_{(x,y)}, C^r_x \) 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].
ND-LTL: Ordered navigation

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

\[(w, i) \models C_x^r \psi \text{ if } 0 < i + r < |w| \text{ and } (w, i + r, d_i|_{x<}) \models \psi,\]
ND-LTL: Ordered navigation

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

- \( (w, i) \models C_x^r \psi \) if \( 0 < i + r < |w| \) and \( (w, i + r, d_i | x < ) \models \psi \),
- \( (w, i, d) \models @x \) if \( d_i | x < = d \),
- \( (w, i, d) \models X^- \psi \) if there is \( j \in pos_d(w) \), \( i < j \), and, for the smallest such \( j \), \( (w, j, d) \models \psi \),
- \( (w, i, d) \models \psi_1 U^- \psi_2 \) if there is \( j \in pos_d(w) \), \( i \leq j \) s.t. \( (w, j, d) \models \psi_2 \) and, for all \( j' \in pos_d(w) \), \( i \leq j' < j \), \( (w, j', d) \models \psi_1 \).
We fix an order on attributes (see examples).
Tuple navigation is restricted to data values of smaller attributes.

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

Fragments of ND-LTL
- \(\text{ND-LTL}^+, \text{ND-LTL}^-\)
Hardness results

**Theorem**

ND-LTL is undecidable.

**Proof.**

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

*ND-LTL is undecidable.*

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

Theorem

*Satisfiability of ND-LTL$^\pm$ is Ackermann-hard.*

Proof.
Using control-state reachability of lossy reset VASS.
Hardness results

Theorem

*Satisfiability of* \( ND-LTL^- \) *over* \( \omega \)-*words is undecidable.*

Proof.

Using *repeated* control-state reachability of lossy reset VASS.
Positive results

**Theorem**

*Satisfiability of \( ND-LTL^- \) over finite words is decidable.*

**Theorem**

*Satisfiability of \( ND-LTL^+ \) is decidable.*

**Proof.**

Use Nested Data Automata
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 \( \Gamma \)
  - 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 \( \text{classes}_i(\mathcal{A}(w)) \subseteq L(\mathcal{B}_i) \).
- pNDA: all states of \( \mathcal{B}_i \) are final
- sNDA: all states of \( \mathcal{B}_i \) are initial
Example

\[
\begin{pmatrix}
  \text{login} & \text{req} & \text{req} & \text{print} & \text{print} & \text{logout} & \text{login} & \ldots \\
  u : & 1 & 1 & 1 & 1 & 1 & 1 & 2 & \ldots \\
\end{pmatrix}
\]

\[
\text{classes}_1(w) = \{\text{login req req print print logout, login} \ldots , \ldots \}
\]

\[
\text{classes}_2(w) = \{\text{request print}\}
\]

\[
L(B_1) = \text{login}(\text{req + print})^* \text{logout}
\]

\[
L(B_2) = \{\text{req print}\}
\]
Theorem

*Emptiness of 2-NDA is undecidable.*
Results

**Theorem**

Emptiness of 2-NDA is undecidable.

**Theorem**

Emptiness of pNDA and \((\omega)\)-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 \( ND-LTL^- \) over finite words is decidable.*

Theorem

*Satisfiability of \( ND-LTL^+ \) 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: $(q_2, \{(q'_1, \{q''_1 : 2, q''_2 : 3\}) : 2, (q'_2, \{q''_2 : 3\}) : 3, (q'_1, \{q''_1 : 1\}) : 1\})$
  - $(q_1, q'_1) \rightarrow (q_2, q'_2, q''_1)$
    - 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)
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 \((a/A) \rightarrow \text{control state move}\)
- guess:
  - \(d_1\) is new: we guess two initial states of \(B_1\) and \(B_2\) which can perform \(A\)
  - \(d_1\) is not new but \(d_2\) is: we take a state in \(B_1\) already remembered and guess an initial state of \(B_2\) and match \(A\)
  - \((d_1, d_2)\) is not new: We match the \(A\) move in both \(B_1\) and \(B_2\)
Conclusion and open problems

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