Logical Investigations on Separation Logics

Day 4: Relationships To Other Logics

Stéphane Demri (CNRS) & Morgan Deters (NYU)

ESSLLI 2015, Barcelona, August 2015
Plan of the talk

- Yesterday’s lecture:
  - Data words and heaps.
  - Arithmetical constraints encoded in 1SL2.
  - Undecidability of 1SL2.
  - Expressive completeness of 1SL2.

- Reduction from data logics.

- Interval temporal logics.

- Modal logics.
Plan of the talk

- Yesterday’s lecture:
  - Data words and heaps.
  - Arithmetical constraints encoded in 1SL2.
  - Undecidability of 1SL2.
  - Expressive completeness of 1SL2.

- Reduction from data logics.

- Interval temporal logics.

- Modal logics.
Data Logics
Separation logic with data

- *kSL* is mainly dedicated to the shape analysis since the equality is the only predicate on the values.
Separation logic with data

- $kSL$ is mainly dedicated to the shape analysis since the equality is the only predicate on the values.

- A sorted list (with a linear ordering):

\[
\forall u, u' \ ((reach(x_1, u) \land reach(u, u') \land reach(u', x_2)) \Rightarrow value(u) \leq value(u'))
\]

\[
\begin{array}{c}
\begin{array}{c}
\text{x}_1 \\
6 \mid l_2 \\
\end{array} \\
\rightarrow \\
\begin{array}{c}
\text{l}_2 \\
22 \mid l_3 \\
\end{array} \\
\rightarrow \\
\begin{array}{c}
\text{x}_2 \\
190 \mid \text{nil} \\
\end{array}
\end{array}
\]
Separation logic with data

- $k$SL is mainly dedicated to the shape analysis since the equality is the only predicate on the values.

- A sorted list (with a linear ordering):

  $\forall u, u' \ ((\text{reach}(x_1, u) \land \text{reach}(u, u') \land \text{reach}(u', x_2)) \Rightarrow value(u) \leq value(u'))$

- One record field (or more) is dedicated to the data domain.

- Pointer arithmetic, sorted lists, data trees, etc....
The logic $kSL[\mathbb{Z}, \sim]$

- Simple example with the data domain $(\mathbb{Z}, \sim)$ but this can be easily generalised to any data domain $(\mathcal{D}, (\mathcal{R}_i)_{i \in I})$.

- Memory state with data $(s, h, \vartheta)$ where
  
  - $(s, h)$ is a memory state for $kSL$.
  
  - $\vartheta$ is a partial function $\mathbb{N} \rightarrow \mathbb{Z}$ so that $\text{dom}(h) = \text{dom}(\vartheta)$.

- $(s, h, \vartheta) \models_f e_1 \sim e_2 \iff \vartheta([e_1])$ and $\vartheta([e_2])$ are defined and $\vartheta([e_1]) = \vartheta([e_2])$.

- ‘$\sim$’ is used instead of ‘$=$’ to avoid any confusion with the equality predicate between locations.

- Next, we show that $1SL3(\ast)[\mathbb{Z}, \sim]$ is undecidable.
A class of \((\alpha, 0)\)-fishbone heaps

\[ \text{dw}'(\alpha, 0) \overset{\text{def}}{=} \text{dw}(\alpha, 0) \land \exists u_0, u_1 \ u_0 \neq u_1 \land \text{mp}(u_0) \land \text{mp}(u_1) \]

\[ \text{mp}(u) \overset{\text{def}}{=} \#u \geq 1 \]
Freeze operator

- Freeze quantifier in hybrid logics.
  
  [Goranko 94; Blackburn & Seligman, JOLLI 95]

- Freeze quantifier in real-time logics.
  
  [Alur & Henzinger, JACM 94]

- Program variable $x$ never decreases below its initial value:
  
  $\exists y (x = y) \land G (x \geq y)$

- Freeze quantifier in imperative programs.
  
  [Manna & Pnueli, 1992]

- Predicate $\lambda$-abstraction [Fitting, JLC 02].
  
  $\langle y \cdot F P (y) \rangle (c)$: current value of constant $c$ satisfies the predicate $P$. 
Freeze operator

- Freeze quantifier in hybrid logics.
  [Goranko 94; Blackburn & Seligman, JOLLI 95]

- Temporal semantics of imperative programs.
  [Manna & Pnueli, 1992]

Program variable $x$ never decreases below its initial value:

$$\exists y \ (x = y) \land G(x \geq y)$$
Freeze operator

- Freeze quantifier in hybrid logics.
  [Goranko 94; Blackburn & Seligman, JOLLI 95]

- Temporal semantics of imperative programs.
  [Manna & Pnueli, 1992]

  Program variable $x$ never decreases below its initial value:

  \[ \exists y \ (x = y) \land G(x \geq y) \]

- Freeze quantifier in real-time logics.
  [Alur & Henzinger, JACM 94]

  \[ y \cdot \varphi(y) \] binds the variable $y$ to the current time $t$. 

- Predicate $\lambda$-abstraction [Fitting, JLC 02].

  \[ \langle y \cdot F P (y) \rangle (c) \] : current value of constant $c$ satisfies the predicate $P$. 

Freeze operator

- Freeze quantifier in hybrid logics.
  [Goranko 94; Blackburn & Seligman, JOLLI 95]

- Temporal semantics of imperative programs.
  [Manna & Pnueli, 1992]
  Program variable $x$ never decreases below its initial value:
  $$\exists y (x = y) \land G(x \geq y)$$

- Freeze quantifier in real-time logics.
  [Alur & Henzinger, JACM 94]
  $y \cdot \varphi(y)$ binds the variable $y$ to the current time $t$.

- Predicate $\lambda$-abstraction [Fitting, JLC 02].
  $\langle y \cdot F P(y) \rangle(c)$: current value of constant $c$ satisfies the predicate $P$. 
Freeze LTL

- LTL$^\alpha_\downarrow(F, F^{-1})$ formulae:

$$\varphi ::= a | \uparrow | \downarrow \varphi | \neg \varphi | \varphi \land \varphi | F \varphi | F^{-1} \varphi$$

(a $\in [1, \alpha]$)

- F and $F^{-1}$ are the usual (non-strict) temporal operators.

- Models: $\omega = (a^1, v^1) \cdots (a^L, v^L) \in ([1, \alpha] \times \mathbb{N})^+$. 

\[ \]
Freeze LTL

- $\text{LTL}_\downarrow^\alpha(F, F^{-1})$ formulae:

  \[ \varphi ::= a \mid \uparrow \mid \downarrow \varphi \mid \neg \varphi \mid \varphi \land \varphi \mid F \varphi \mid F^{-1} \varphi \]

  \[ (a \in [1, \alpha]) \]

- $F$ and $F^{-1}$ are the usual (non-strict) temporal operators.

- Models: $\omega = (a^1, v^1) \cdots (a^L, v^L) \in ([1, \alpha] \times \mathbb{N})^+$.

- $\downarrow \varphi$: to store the current data value and then evaluate $\varphi$.

- $\uparrow$: equality test between the stored value and the current data value.

- Relation $\models_v$ parameterised by the stored data value.
Satisfaction relation

\[ \varrho w = (a^1, v^1) \cdots (a^L, v^L) \in ([1, \alpha] \times \mathbb{N})^+ \]

\[ \varrho w, i \models_v a \qquad \overset{\text{def}}{\iff} \qquad a^i = a \]

\[ \varrho w, i \models_v \uparrow \qquad \overset{\text{def}}{\iff} \qquad v^i = v \]

\[ \varrho w, i \models_v \varphi \qquad \overset{\text{def}}{\iff} \qquad \varrho w, i \models_{v^i} \varphi \]

\[ \varrho w, i \models_v F \varphi \qquad \overset{\text{def}}{\iff} \qquad \text{there is } i' \in [i, L] \text{ such that } \varrho w, i' \models_{v^i} \varphi \]

\[ \varrho w, i \models_v F^{-1} \varphi \qquad \overset{\text{def}}{\iff} \qquad \text{there is } i' \in [1, i] \text{ such that } \varrho w, i' \models_{v^i} \varphi \]

\[ G(a \Rightarrow \downarrow G(b \Rightarrow \neg \uparrow)) \]
Satisfaction relation

\[ \partial w = (a^1, v^1) \cdots (a^L, v^L) \in ([1, \alpha] \times \mathbb{N})^+ \]

\[ \partial w, i \models_v a \iff a^i = a \]

\[ \partial w, i \models_v \uparrow \iff v^i = v \]

\[ \partial w, i \models_v \downarrow \varphi \iff \partial w, i \models_{v^i} \varphi \]

\[ \partial w, i \models_v F\varphi \iff \text{there is } i' \in [i, L] \text{ such that } \partial w, i' \models_{v^i} \varphi \]

\[ \partial w, i \models_v F^{-1}\varphi \iff \text{there is } i' \in [1, i] \text{ such that } \partial w, i' \models_{v^i} \varphi \]

\[ G(a \Rightarrow \downarrow G(b \Rightarrow \neg \uparrow)) \]

The satisfiability problem for \( \text{LTL}^\alpha_{\downarrow}(F, F^{-1}) \) is undecidable.

[Figueira & Segoufin, MFCS’09]
Principles of the translation

\[ T(\varphi) \overset{\text{def}}{=} \text{dw}'(\alpha, 0) \land tr(u_0, \varphi) \land mp(u_0) \land \neg \exists u_1 \ (u_1 \leftrightarrow u_0 \land mp(u_1)) \]

- “mp(u_0)”: \(u_0\) on the main path.

\[ \text{dw}'(\alpha, 0) \overset{\text{def}}{=} \text{dw}(\alpha, 0) \land \exists u_0, u_1 \ u_0 \neq u_1 \land mp(u_0) \land mp(u_1) \]

- \(tr\) simply internalises the semantics of \(\text{LTL}_\downarrow^\alpha(F, F^{-1})\).

- \(\downarrow\) is an existential quantification and \(\uparrow\) is an equality test.

- \(u_2\) is used to simulate \(\uparrow\) and \(\downarrow\).
Translation

\[ tr(u_i, \uparrow) \overset{\text{def}}{=} (u_2 \sim u_i) \]

\[ tr(u_i, a) \overset{\text{def}}{=} (\#u_i = a + 2) \]

\[ tr(u_i, \downarrow \psi) \overset{\text{def}}{=} \exists u_2 \ ((u_2 \sim u_i) \land tr(u_i, \psi)) \]

\[ tr(u_i, F\psi) \overset{\text{def}}{=} \exists u_{1-i} \ (tr(u_{1-i}, \psi) \land mp(u_{1-i}) \land \neg last(u_{1-i}) \land reach(u_i, u_{1-i})) \]

\[ tr(u_i, F^{-1}\psi) \overset{\text{def}}{=} \exists u_{1-i} \ (tr(u_{1-i}, \psi) \land mp(u_{1-i}) \land \neg last(u_{1-i}) \land reach(u_{1-i}, u_i)) \]

\((\#u_i = a + 2), \ reach(u_i, u_{1-i}), \ mp(u_{1-i}), \ last(u_{1-i}), \ dw'(\alpha, 0)\) definable in 1SL2(\(\ast\)).
Correctness

- \( \varphi \) is \( \text{LTL}^{\alpha}(F, F^{-1}) \) satisfiable iff \( T(\varphi) \) is \( 1\text{SL3}(\ast)[\mathbb{Z}, \sim] \) satisfiable.

- Easy to add \( X \) and \( X^{-1} \).

- The satisfiability problem for \( 1\text{SL3}(\ast)[\mathbb{Z}, \sim] \) is undecidable.  
  \[ \text{[Bansal \& Brochenin \& Lozes, FOSSACS’09]} \]

- Decidable fragments with short-distance comparisons can be found in \[ \text{[Bansal \& Brochenin \& Lozes, FOSSACS’09].} \]

- Other prominent logic \( \text{STRAND} \) stating properties on heap structures and the data. \[ \text{[Madhusudan \& Parlato \& Qiu, POPL’11]} \]
Encoding data values in the heap structure

- Reduction from a temporal logic on data words into a separation logic with data.

- Next, a reduction from a first-order logic on data words into a separation logic (without data).

- Again, data values are encoded by numbers of predecessors.

- Undecidability of 1SL2 can be also shown this way.
First-order logic on data words

- **FO2_{\alpha,\beta}(\prec, +1, =, \sim, \precj) formulae:**

  \[ \varphi ::= a(v) \mid v \sim j v \mid v \precj v \mid v < v \mid v = 1 + (v) \mid v = v \mid \neg \varphi \mid \varphi \land \varphi \mid \exists v \varphi \]

  where \( a \in [1, \alpha], j \in [1, \beta] \) and \( v ::= u_1 \mid u_2 \).

- Quantifications are over positions.

- \( \sim j \) and \( \precj \) about the \( j \)th data value.
Semantics

\[ \mathcal{dw} = (a^1, v^1_1, \ldots, v^1_\beta) \cdots (a^L, v^L_1, \ldots, v^L_\beta) \]

\[ f : \{u_1, u_2\} \rightarrow [1, L] \]

\[ \mathcal{dw} \models_f a(u_i) \quad \Leftrightarrow \quad a^{f(u_i)} = a \]
\[ \mathcal{dw} \models_f u_i \sim_j u_i' \quad \Leftrightarrow \quad v^{f(u_i)}_j = v^{f(u_i')}_j \]
\[ \mathcal{dw} \models_f u_i \prec_j u_i' \quad \Leftrightarrow \quad v^{f(u_i)}_j < v^{f(u_i')}_j \]
\[ \mathcal{dw} \models_f u_i = u_i' \quad \Leftrightarrow \quad f(u_i) = f(u_i') \]
\[ \mathcal{dw} \models_f u_i = 1 + (u_i') \quad \Leftrightarrow \quad f(u_i) = f(u_i') + 1 \]
\[ \mathcal{dw} \models_f u_i < u_i' \quad \Leftrightarrow \quad f(u_i) < f(u_i') \]
\[ \mathcal{dw} \models_f \exists u_i \varphi \quad \Leftrightarrow \quad \text{there is } p \in [1, L] \text{ s.t. } \mathcal{dw} \models_{f[u_i \mapsto p]} \varphi \]
Results about FO2 on data words

- $\bigcup_{\alpha \geq 1} \text{FO2}_{\alpha,0}(<, +1, =)$: \text{NExpTime}-complete.  
  [Etessami & Vardi & Wilke, LICS’97]

- $\bigcup_{\alpha \geq 1} \text{FO2}_{\alpha,1}(<, +1, =, \sim)$: decidable.

- $\bigcup_{\alpha \geq 1} \text{FO2}_{\alpha,2}(<, +1, =, \sim)$: undecidable.

- $\bigcup_{\alpha \geq 1} \text{FO2}_{\alpha,1}(<, +1, =, \sim, \prec)$: undecidable.  
  [Bojańczyk et al., LICS’06]
Position: every two locations
Reduction from $\bigcup_{\alpha \geq 1} \text{FO2}_{\alpha,1}(<, +1, =, \sim, \prec)$ to 1SL2

\[
\begin{align*}
tr(u_i = u_j) & \overset{\text{def}}{=} u_i = u_j \\
tr(u_i < u_j) & \overset{\text{def}}{=} \text{reach}(u_i, u_j) \land u_i \neq u_j \\
tr(u_j = 1 + (u_i)) & \overset{\text{def}}{=} \top \land (\text{sreach}(u_i, u_j) \land (\#u_i^{+2} = 1) \land \neg(\#u_i^{+3} \geq 0)) \\
tr(a(u_i)) & \overset{\text{def}}{=} \exists u_{3-i} \ (u_{3-i} \leftrightarrow u_i) \land (\#u_{3-i} = a + 2) \\
tr(u_i \sim_1 u_j) & \overset{\text{def}}{=} \#u_i = \#u_j \\
tr(u_i \prec_1 u_j) & \overset{\text{def}}{=} \#u_i + 1 \leq \#u_j \\
tr(\exists u_i \varphi) & \overset{\text{def}}{=} \exists u_i \ (\#u_i \geq \alpha + 3) \land tr(\varphi)
\end{align*}
\]

\text{sreach}(u_i, u_j): \text{the heap is only made of a path from } u_i \text{ to } u_j \\
(\#u_i^{+3} = 1): \text{the 3rd location after } u \text{ has one predecessor.} \\
\#u_i = \#u_j: u_i \text{ and } u_j \text{ have the same amount of predecessors.}
Recapitulation

- The satisfiability problem for 1SL2 is undecidable.

- Undecidability results for separation logics can be obtained by reduction from data logics.

- Next, lower bound complexity for 1SL2(*) can be obtained by reduction from interval temporal logics.
Interval Temporal Logics
Point-based vs. interval-based temporal logics

- Temporal logics CTL, CTL*, ECTL, ... are point-based.
- Reasoning about intervals. [Allen, CACM 83]
- Propositional Interval Temporal Logic (PITL). [Moszkowski, PhD 83]
- Decidability with the locality condition.
- Undecidability is very common in the realm of interval-based temporal logics. [Bresolin et al., AMAI 14]
PITL

- $\Sigma = [1, \alpha]$.

- Models are non-empty finite words on $\Sigma$. (data words with $\beta = 0$)

- $\varphi ::= a \mid pt \mid \neg \varphi \mid \varphi \land \varphi \mid \varphi \lor \varphi$

- Chop (with overlap):
  \[
  \text{chops} \overset{\text{def}}{=} \{(w_1, w_2, w_3) \in (\Sigma^+)^3 \mid \exists a, w', w'' \text{ such that } w_1 = w' a w'', w_2 = w' a, w_3 = a w''\}
  \]

- $(bba, bb, ba) \in \text{chops}$. 
Semantics for PITL

\( w \models a \quad \overset{\text{def}}{\iff} \quad \text{the first letter of } w \text{ is } a \)

\( w \models pt \quad \overset{\text{def}}{\iff} \quad \text{the length of } w = 1 \)

\( w \models \neg \varphi \quad \overset{\text{def}}{\iff} \quad w \not\models \varphi \)

\( w \models \varphi \land \psi \quad \overset{\text{def}}{\iff} \quad w \models \varphi \text{ and } w \models \psi \)

\( w \models \varphi \text{ C } \psi \quad \overset{\text{def}}{\iff} \quad \text{there exist words } w_1, w_2 \text{ such that } \text{chops} (w, w_1, w_2), w_1 \models \varphi \text{ and } w_2 \models \psi \)

\( \triangleright \quad \text{bab } \models (b \text{ C } a) \text{ C } \neg pt \text{ with } \Sigma = \{a, b\}. \)

\( (ba \models b \text{ C } a \text{ and } ab \models \neg pt.) \)

\( \triangleright \quad \text{pt } \land (a \text{ C } b) \text{ is unsatisfiable.} \)
Decidability and complexity

- Given $\alpha \geq 1$ and $\Sigma = [1, \alpha]$, the problem $\text{SAT}(\text{PITL}_\Sigma)$ is decidable, but with $\alpha \geq 2$ is not elementary recursive.

  [Halpern, Kozen, Moszkowski, 80's]

- Chop in PITL can be encoded with separating conjunction in 1SL2($\ast$).

- The satisfiability problem for 1SL2($\ast$) is decidable but not elementary recursive.

  [Demri & Deters, TOCL 15]
Decidability and complexity

- Given $\alpha \geq 1$ and $\Sigma = [1, \alpha]$, the problem SAT(PITL$_\Sigma$) is decidable, but with $\alpha \geq 2$ is not elementary recursive.  
  \[ \text{[Halpern, Kozen, Moszkowski, 80's]} \]

- Chop in PITL can be encoded with separating conjunction in 1SL2($\ast$).

- The satisfiability problem for 1SL2($\ast$) is decidable but not elementary recursive.  
  \[ \text{[Demri & Deters, TOCL 15]} \]
Concurrent Separation Logic – Temporal Separation

Tony Hoare, Microsoft Research, Cambridge
Peter O’Hearn, Queen Mary, University of London

Separation Logic already has a notion of spatial separation.

Hoare and O’Hearn consider separation in time in the article Separation Logic Semantics for Communicating Processes.

They use state sequences and sequential operator similar to ITL’s.

Quote from Zhou Chaochen in recent book:

Moreover, in a personal communication, Tony Hoare drew my attention to the possible correspondence between the chop operator in Interval Logic and the separating (spatial) conjunction in Separation Logic. I believe that Interval Logic deserves an important role in computing science.
A correspondence between words and heaps

\[ \sim \overset{\text{def}}{=} \{(w, h_w) \mid w \in \Sigma^+\} \cup \{(\epsilon, \emptyset)\} \]

- \( h_w \): \((\alpha, 0)\)-fishbone heap corresponding to \( w \in [1, \alpha]^+ \).

- \( \sim \) is a bijection between \([1, \alpha]^*\) and the set of (equivalence classes of isomorphic) \((\alpha, 0)\)-fishbone heaps augmented with the empty heap.

- Every word \( w \) is in \( \text{dom}(\sim) \).

- Every \((\alpha, 0)\)-fishbone heap is in \( \text{ran}(\sim) \).

- If \( w \sim h \), \( h \) is either empty or an \((\alpha, 0)\)-fishbone heap.

- If \( w \sim h \), then \( w \) is empty iff \( \text{dom}(h) \) is empty.
Clean cut

- A clean cut neatly cleaves a heap representation of a word into two subheaps in correspondence with two subwords.

- A clean cut \((h_1, h_2)\) for the \((\alpha, 0)\)-fishbone heap \(h\):
  - \(h = h_1 \cup h_2\).
  - There are words \(w_1\) and \(w_2\) such that \(w_1 \sim h_1\) and \(w_2 \sim h_2\).
  - \(w_1 w_2 \sim h\).
Two cuts on 2 4 1 5 2 5
Properties about clean cuts

- Let $w \sim h$ with $w = w_1w_2 \in \Sigma^*$. There exist $h_1$ and $h_2$ such that $h = h_1 \uplus h_2$, $w_1 \sim h_1$, and $w_2 \sim h_2$.

- When $w_1$ and $w_2$ are non-empty, the proof is obtained by defining a clean cut on $h$.

- $(h_1, h_2)$ is clean cut of the fishbone heap $h$ whenever:
  - $w \sim h$ and $f(u)$ is on the main path of $h$,
  - $h \models f\text{alloc}(u)$,
  - $h_1 \models f\text{dw}(a, 0) \land \neg\text{alloc}(u)$,
  - $h_2 \models f\text{dw}(a, 0) \land \#u = 0$. 

The ghost letter

- The correspondence between the chop operation and separating conjunction is not an exact one.

- Ghost letter: internal letter carried by the translation.

- Translation from PITL into 1SL2(\ast) parameterised by the ghost letter.

- To represent \(c w\), we use \(h \sim w\) with the ghost letter \(c\).

- Given \(aw = a w' b w'' \& \:
\[ w \sim h, \text{ there are } h_1, h_2 \text{ s.t.:} \]
  - \(h = h_1 \cup h_2\).
  - \(w' b \sim h_1 \text{ and } w'' \sim h_2\).
Principles of the translation

\[ t(\varphi) \overset{\text{def}}{=} (d\omega(\alpha, 0) \lor \text{emp}) \land \bigvee_{a \in \Sigma} t_a(\varphi) \]

Three types of chopping:

<table>
<thead>
<tr>
<th>( w )</th>
<th>ghost letter 1</th>
<th>subheap 1</th>
<th>ghost letter 2</th>
<th>subheap 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>( aw_1 b w_2 )</td>
<td>( a )</td>
<td>( w_1 b \sim h_1 )</td>
<td>( b )</td>
<td>( w_2 \sim h_2 )</td>
</tr>
<tr>
<td>( aw_1 b )</td>
<td>( a )</td>
<td>( w_1 b \sim h_1 )</td>
<td>( b )</td>
<td>( \epsilon \sim h_2 )</td>
</tr>
<tr>
<td>( aw_2 )</td>
<td>( a )</td>
<td>( \epsilon \sim h_1 )</td>
<td>( a )</td>
<td>( w_2 \sim h_2 )</td>
</tr>
</tbody>
</table>
Translation

\[ t_a(b) \overset{\text{def}}{=} \top \text{ if } b = a \]

\[ t_a(b) \overset{\text{def}}{=} \bot \text{ if } b \neq a \]

\[ t_a(\text{pt}) \overset{\text{def}}{=} \text{emp} \]

\[ t_a(\varphi \mathbf{C} \psi) \overset{\text{def}}{=} \text{chop1}_a \lor \text{chop2}_a \lor \text{chop3}_a \]

\[ \text{chop1}_a \overset{\text{def}}{=} \bigvee_{b \in \Sigma} \exists u (\#u = b + 2 \land \left[ \text{dw}(\alpha, 0) \land \neg \text{alloc}(u) \land t_a(\varphi) \ast \ldots \ldots \text{dw}(\alpha, 0) \land \#u = 0 \land t_b(\psi) \right] \}

\[ \text{chop2}_a \overset{\text{def}}{=} \bigvee_{b \in \Sigma} (\exists u \text{last}(u) \land \#u = b + 2) \land \ldots \ldots [t_a(\varphi) \ast (\text{emp} \land t_b(\psi))] \]

\[ \text{chop3}_a \overset{\text{def}}{=} (\text{emp} \land t_a(\varphi)) \ast t_a(\psi) \]
Correctness

- Translation parameterised by the alphabet $\Sigma = [1, \alpha]$.
- The translation produces formulae in $1\text{SL}2(\ast)$.
- $t_a(\varphi)$ is a closed formula.
- Exponential blow-up as soon as $\alpha \geq 2$.
- Given $w \sim h$, for every $\varphi$, we have $aw \models \varphi$ iff $h \models t_a(\varphi)$. 
Non-elementarity

- $\varphi$ is satisfiable iff the 1SL2($\ast$) formula $t(\varphi)$ is satisfiable.

- The satisfiability problem for 1SL2($\ast$) is decidable but not elementary recursive.
  - $\text{PITL}_\Sigma$ is not elementary recursive when $\alpha \geq 2$.  
    [Moszkowski, PhD83]

- 1SL($\ast$) is decidable.  
  [Rabin69, TAMS 69; Brochenin & Demri & Lozes, IC 12]

- Alternative proof by reduction from the nonemptiness problem for regular expressions over $\{0, 1\}$ with union, concatenation and complement.  
  [Stockmeyer, PhD 74]
Modal Logics
Modal logic for heaps (MLH)

- Model for MLH $M: (\mathbb{N}, R)$ such that $R$ is a finite and functional.

- MLH models are heap graphs with one record field.

- Formulae:

$$\varphi ::= \perp \mid \neg \varphi \mid \varphi \land \varphi \mid \Diamond \varphi \mid \Diamond^{-1} \varphi \mid \langle \neq \rangle \varphi \mid \langle \ast \rangle \varphi \mid \varphi \ast \varphi \mid \varphi \rightarrow \ast \varphi.$$ 

- No atomic propositions as in Hennessy-Milner logic HML.

[Hennessy & Milner, ICALP’80]
Satisfaction relation

\( M, l \models \diamond \varphi \) \iff \text{there is } l' \text{ s.t. } (l, l') \in R \text{ and } M, l' \models \varphi

\( M, l \models \diamond^{-1} \varphi \) \iff \text{there is } l' \text{ s.t. } (l', l) \in R \text{ and } M, l' \models \varphi

\( M, l \models \langle \star \rangle \varphi \) \iff \text{there is } l' \text{ s.t. } (l, l') \in R^* \text{ and } M, l' \models \varphi

\( M, l \models \langle \neq \rangle \varphi \) \iff \text{there is } l' \neq l \text{ s.t. } M, l' \models \varphi
Satisfaction relation

\( M, l \models \Diamond \varphi \quad \iff \quad \text{there is } l' \text{ s.t. } (l, l') \in R \text{ and } M, l' \models \varphi \)

\( M, l \models \Diamond^{-1} \varphi \quad \iff \quad \text{there is } l' \text{ s.t. } (l', l) \in R \text{ and } M, l' \models \varphi \)

\( M, l \models \langle * \rangle \varphi \quad \iff \quad \text{there is } l' \text{ s.t. } (l, l') \in R^* \text{ and } M, l' \models \varphi \)

\( M, l \models \langle \neq \rangle \varphi \quad \iff \quad \text{there is } l' \neq l \text{ s.t. } M, l' \models \varphi \)

\( M, l \models \varphi_1 \ast \varphi_2 \quad \iff \quad \text{for some partition } \{R_1, R_2\} \text{ of } R \)
\( (N, R_1), l \models \varphi_1 \text{ and } (N, R_2), l \models \varphi_2 \)

\( M, l \models \varphi_1 \ast \varphi_2 \quad \iff \quad \text{for all models } M' = (N, R') \text{ s.t. } R \cap R' = \emptyset \)
\( \text{and } R \cup R' \text{ is functional, } M', l \models \varphi_1 \text{ implies } (N, R \cup R'), l \models \varphi_2 \)
Abbreviations

\[\langle U \rangle \varphi \quad \text{def} \quad \varphi \lor \langle \neq \rangle \varphi\]

\[\lbrack U \rbrack \varphi \quad \text{def} \quad \neg \langle U \rangle \neg \varphi\]

\[\lbrack \neq \rbrack \varphi \quad \text{def} \quad \neg \langle \neq \rangle \neg \varphi\]

\[\Diamond \geq_k \top \quad \text{def} \quad \Diamond^{-1} \top \ast \ldots \ast \Diamond^{-1} \top \ (k \geq 1 \text{ times})\]

\[\Diamond \leq_k \top \quad \text{def} \quad \neg \Diamond \geq_k \top\]

\[\Diamond \leq_k \top \quad \text{def} \quad \Diamond \geq_{k_1} \top \land \Diamond \leq_{k_2} \top\]

\[\Diamond =_k \top \quad \text{def} \quad \Diamond \geq_k \top \land \Diamond \leq_k \top\]
Relational translation into 1SL2

$tr$ is homomorphic for Boolean and separating connectives.

$\begin{align*}
tr(\bot, u_i) & \overset{\text{def}}{=} \bot \\
tr(\Diamond \varphi, u_i) & \overset{\text{def}}{=} \exists u_{3-i} (u_i \rightarrow u_{3-i}) \land tr(\varphi, u_{3-i}) \\
tr(\Diamond^{-1} \varphi, u_i) & \overset{\text{def}}{=} \exists u_{3-i} (u_{3-i} \rightarrow u_i) \land tr(\varphi, u_{3-i}) \\
tr(\langle \neq \rangle \varphi, u_i) & \overset{\text{def}}{=} \exists u_{3-i} (u_i \neq u_{3-i}) \land tr(\varphi, u_{3-i}) \\
tr(\langle \star \rangle \varphi, u_i) & \overset{\text{def}}{=} \exists u_{3-i} \text{ reach}(u_i, u_{3-i}) \land tr(\varphi, u_{3-i}).
\end{align*}$
Correctness

- \( \varphi \) in MLH is satisfiable iff \( \exists u_1 \ tr(\varphi, u_1) \) is satisfiable in 1SL2.

- If \( \varphi \) is in MLH(\(*\)), then \( \exists u_1 \ tr(\varphi, u_1) \) is in 1SL2(\(*\)).

- The proof is similar to the proof for the translation from the modal logic K into FO2.

- Formulae in 1SL2 can be interpreted over MLH models.

- By structural induction, \( M, I \models \psi \) iff \( M \models_{[u_i \mapsto I]} tr(\psi, u_i) \).

- The satisfiability problem for MLH(\(*\)) is decidable.
MLH fragments/extension

- MLH(∗) can be viewed as a fragment of 1SL2(∗).
- ⟨≠⟩⁻¹ and ⟨∗⟩⁻¹ can be added at no cost.
- Open problem: decidability status of MLH.
- Next, we show that MLH(∗) is non-elementary recursive.
Fishbone heaps (see Lecture 3)

- (fb1): $\text{dom}(h) \neq \emptyset$.
- (fb2): There is a location reachable from all the locations of $\text{dom}(h)$ that is not in $\text{dom}(h)$.
- (fb3): The pattern $l_1 \rightarrow l_2 \rightarrow l_3 \leftarrow l_4 \leftarrow l_5$ is forbidden.
Fishbone heaps (see Lecture 3)

- (fb1): $\text{dom}(h) \neq \emptyset$.

- (fb2): There is a location reachable from all the locations of $\text{dom}(h)$ that is not in $\text{dom}(h)$.

- (fb3): The pattern $l_1 \rightarrow l_2 \rightarrow l_3 \leftarrow l_4 \leftarrow l_5$ is forbidden.

\[
\varphi_{fb} \overset{\text{def}}{=} \left( \langle U \rangle \diamond T \right) \land \left( \neg \langle U \rangle (\diamond^{-1} \diamond^{-1} T \cdot \diamond^{-1} \diamond^{-1} T) \right) \land \\
\langle U \rangle ((\diamond^{-1} T \land \neg \diamond T) \land \not\equiv (\diamond^{-1} T \land \neg \diamond T)) \land [U](\diamond T \Rightarrow \langle \ast \rangle (\diamond^{-1} T \land \neg \diamond T))
\]
\((\alpha, \beta)\)-fishbone heaps

- (C1): The first location on the main path has a number of predecessors in \([3, \alpha + 2]\).

- (C2): On the main path, a location with a number of predecessors in \([3, \alpha + 2]\), is followed by \(\beta\) locations with at least \(\alpha + 3\) predecessors.

- (C3): The number of locations on the main path is a multiple of \(\beta + 1\).
\((\alpha, \beta)\)-fishbone heaps

- **(C1):** The first location on the main path has a number of predecessors in \([3, \alpha + 2]\).

- **(C2):** On the main path, a location with a number of predecessors in \([3, \alpha + 2]\), is followed by \(\beta\) locations with at least \(\alpha + 3\) predecessors.

- **(C3):** The number of locations on the main path is a multiple of \(\beta + 1\).

\[
\varphi_{(C1)} \overset{\text{def}}{=} \langle U \rangle((\Diamond^{-1} T) \land (\neg \Diamond^{-1} \Diamond^{-1} T) \land \Diamond^{-1} [3,\alpha+2] T)
\]

\[
\varphi_{(C2)} \overset{\text{def}}{=} [U](\Diamond^{-1} [3,\alpha+2] T \Rightarrow \bigwedge_{i\in[1,\beta]} i \text{ times } (\Diamond \cdots \Diamond \Diamond^{-1} \geq_{\alpha+3} T))
\]

\[
\varphi_{(C3)} \overset{\text{def}}{=} [U](\Diamond^{-1} [3,\alpha+2] T \Rightarrow (\neg (\Diamond \cdots \Diamond T) \lor \Diamond \cdots \Diamond (\Diamond^{-1} [3,\alpha+3] T)))
\]
Characterisation

\[ dw^\Box (\alpha, \beta) \overset{\text{def}}{=} \varphi_{fb} \land \varphi_{(C1)} \land \varphi_{(C2)} \land \varphi_{(C3)} \]

- \( h \models dw^\Box (\alpha, \beta) \) iff \( h \) is an \((\alpha, \beta)\)-fishbone heap.

- \( dw^\Box (\alpha, \beta) \) belongs to \( \text{MLH}(\ast) \).

- The translation from PITL to \( \text{MLH}(\ast) \) uses the same principles as for \( 1\text{SL2}(\ast) \).

\[ t(\varphi)\Box \overset{\text{def}}{=} (dw^\Box (\alpha, 0) \lor ([U]\neg \Diamond \top)) \land \left( \bigvee_{a \in \Sigma} t(\varphi)^\Box_a \right) \]
Translation into MLH

- \( t(a) \uparrow_a \overset{\text{def}}{=} \top \) and \( t(b) \uparrow_a \overset{\text{def}}{=} \bot \) for every letter \( b \in \Sigma \setminus \{a\} \).

- \( t(pt) \uparrow_a \overset{\text{def}}{=} ([U] \neg \Diamond \top) \).

- \( t(\varphi \mathbf{C} \psi) \uparrow_a \overset{\text{def}}{=} \text{chop1}_a \lor \text{chop2}_a \lor \text{chop3}_a \) with

  \[
  \text{chop1}_a \overset{\text{def}}{=} \bigvee_{b \in \Sigma} \langle U \rangle ((\Diamond_{=b+2}^{-1} \top \land \text{dw}_\square(\alpha, 0) \land \neg \Diamond \top \land t(\varphi) \uparrow_a) \ast \ldots \ldots (\text{dw}_\square(\alpha, 0) \land \neg \Diamond_{=b+2}^{-1} \top \land t(\psi) \uparrow_b))
  \]

  \[
  \text{chop2}_a \overset{\text{def}}{=} (\bigvee_{b \in \Sigma} \langle U \rangle ((\Diamond_{=b+2}^{-1} \top \land \neg \Diamond \top) \land \Diamond_{=b+2}^{-1} \top) \land (t(\varphi) \uparrow_a \ast \ldots \ldots (t(\psi) \uparrow_b \land ([U] \neg \Diamond \top)))))
  \]

  \[
  \text{chop3}_a \overset{\text{def}}{=} (((t(\varphi) \uparrow_a \land ([U] \neg \Diamond \top)) \ast t(\psi) \uparrow_a))
  \]
Correctness

- Translation parameterised by the alphabet $\Sigma = [1, \alpha]$.
- The translation produces formulae in $\text{MLH}(*)$.
- Exponential blow-up as soon as $\alpha \geq 2$.
- $\varphi$ is satisfiable iff the $\text{MLH}(*)$ formula $t(\varphi) \square$ is satisfiable.
- The satisfiability problem for $\text{MLH}(*)$ is decidable but not elementary recursive.
Decidability of 1MSOL

- Decidability of the weak monadic second-order theory on $(\mathcal{D}, f, =)$ where $\mathcal{D}$ is a countable domain, $f$ is a unary function, and $=$ is equality.  
  
  [Rabin, TAMS 69]

- Weakness means quantifications over finite sets.

- 1MSOL is decidable by translation into that theory.  
  
  [Brochenin & Demri & Lozes, IC 12]

- Decidability of 1SL(∗) by translation into 1MSOL
\[
\exists P \ (\forall u \ P(u) \Leftrightarrow (\exists u' \ u \rightarrow u')) \land tr_P(\varphi)
\]

\[
tr_P(u \rightarrow u') \quad \overset{\text{def}}{=} \quad P(u) \land u \rightarrow u'
\]

\[
tr_P(u = u') \quad \overset{\text{def}}{=} \quad u = u'
\]

\[
tr_P(\varphi \ast \psi) \quad \overset{\text{def}}{=} \quad \exists Q, Q' \ (P = Q \uplus Q') \land tr_Q(\varphi) \land tr_{Q'}(\psi)
\]

\[
(P = Q \uplus Q') \overset{\text{def}}{=} \forall u \ (P(u) \Leftrightarrow (Q(u) \lor Q'(u))) \land \neg (Q(u) \land Q'(u))
\]
Expressive power

- $1SL(\ast)$ is strictly less expressive than $1MSOL$.
  [Antonopoulos & Dawar, FOSSACS’09]

- The proof uses Ehrenfeucht-Fraïssé games.

- No formula in $1SL(\ast)$ characterises the forests of binary trees such that there is one binary tree whose number of leaves is a multiple of 3.
Property in 1MSOL

\[ \forall u \left( \# u \leq 2 \wedge \exists \bar{u} \left( \text{reach}(u, \bar{u}) \wedge \neg \text{alloc}(\bar{u}) \right) \right). \]

\[ \exists u \left( \neg \text{alloc}(u) \wedge \# u \geq 1 \right) \wedge \]
\[ \exists P, P_0, P_1, P_2 \left( \left( \forall \bar{u} P(\bar{u}) \iff \text{reach}(\bar{u}, u) \right) \wedge \left( P = P_0 \cup P_1 \cup P_2 \right) \wedge P_0(u) \wedge \right. \]
\[ \left. \left( \forall \bar{u} \left( P(\bar{u}) \wedge \# \bar{u} = 0 \right) \Rightarrow P_1(\bar{u}) \right) \right) \wedge \]
\[ \left( \forall \bar{u}, \bar{u}' \left( \bar{u} \leftrightarrow \bar{u}' \right) \wedge P(\bar{u}) \wedge \# \bar{u}' = 1 \right) \Rightarrow \bigwedge_{i=0}^{2} \left( P_i(\bar{u}) \iff P_i(\bar{u}') \right) \right) \wedge \]
\[ \bigwedge_{i,j,k \in \{0,1,2\}, i \equiv 3j+k} \left( \forall \bar{u}, \bar{u}', \bar{u}'' \left( P_j(\bar{u}') \wedge P_k(\bar{u}'') \wedge (\bar{u}' \neq \bar{u}'') \wedge (\bar{u}' \leftrightarrow \bar{u}) \wedge \right. \right. \]
\[ \left. \left. (\bar{u}'' \leftrightarrow \bar{u}) \Rightarrow P_i(\bar{u}) \right) \right) \]
Conclusion

Today’s lecture:

- Undecidability proofs by reduction from data logics.
- Non-elementarity proof by reduction from propositional interval temporal logic.
- A modal logic of heap.

Tomorrow’s lecture:

- SMT framework and translations.
- Expressiveness of $k$SL0 and PACE upper bound.
- Translations into first-order logic and QBF.
- Concluding remarks about the course.
Conclusion

Today’s lecture:

- Undecidability proofs by reduction from data logics.
- Non-elementarity proof by reduction from propositional interval temporal logic.
- A modal logic of heap.

Tomorrow’s lecture:

- SMT framework and translations.
- Expressiveness of $kSL0$ and $PSPACE$ upper bound.
- Translations into first-order logic and QBF.
- Concluding remarks about the course.