Logical Investigations on Separation Logics

Day 3: Expressiveness of Separation Logics

Stéphane Demri (CNRS) & Morgan Deters (NYU)

ESSLLI 2015, Barcelona, August 2015
Plan of the talk

▶ Yesterday’s lecture:
  ▶ PSPACE-hardness proofs for $k$SL0 fragments.
  ▶ Main complexity results about $k$SL0 fragments.
  ▶ Encoding runs of Minsky machines in abstract propositional separation logics.
Plan of the talk

- Yesterday’s lecture:
  - PSPACE-hardness proofs for $k$SL0 fragments.
  - Main complexity results about $k$SL0 fragments.
  - Encoding runs of Minsky machines in abstract propositional separation logics.

- Data words and heaps.

- Arithmetical constraints encoded in 1SL2.

- Undecidability of 1SL2.

- Expressive completeness of 1SL.
Encoding Data Words by Heaps
Ubiquity of data words [Bouyer, IPL 02]

- Data word: $a_1 \ a_2 \ a_3 \ \cdots$
  $v_1 \ v_2 \ v_3 \ \cdots$
  - Each $a_i$ belongs to a finite alphabet $\Sigma$.
  - Each $v_i$ belongs to an infinite domain $D$. 
Ubiquity of data words [Bouyer, IPL 02]

- Data word: $a_1 \ a_2 \ a_3 \ \cdots$
  
  $
u_1 \ \nu_2 \ \nu_3 \ \cdots$
  
  - Each $a_i$ belongs to a finite alphabet $\Sigma$.
  - Each $\nu_i$ belongs to an infinite domain $D$.

- Timed word [Alur & Dill, TCS 94]
  
  $a \ b \ c \ a \ a \ b$
  
  $0 \ 0.3 \ 1 \ 2.3 \ 3.5 \ 3.51$

- Runs from counter systems
  
  $q_0 \ q_2 \ q_3 \ q_2 \ q_3 \ q_2$
  
  $0 \ 0 \ 1 \ 2 \ 3 \ 4$

- Integer arrays [Habermehl & Iosif & Vojnar, FOSSACS’08]
  

Explicit relationships with data words and related formalisms in [Alur & Černý & Weinstein, CSL’09].
Finite alphabet and infinite domain
Specifying classes of data words

- Register automata
  - Register automata
  - Data automata
  - Survey
  - Class automata

- First-order languages

- Temporal logics
  - Real-time logic TPTL
  - LTL with registers

- Many other formalisms
  - Rewriting systems with data
  - Hybrid logics
  - ...
Data words with multiple attributes

- $(a^1, v^1_1, \ldots, v^1_\beta) \cdots (a^L, v^L_1, \ldots, v^L_\beta) \in ([1, \alpha] \times \mathbb{N}^\beta)^+$

- $\omega_0 = (4, 1)(3, 2)(4, 2) \in ([1, 4] \times \mathbb{N})^+.$
Data words with multiple attributes

▶ \((a^1, v^1_1, \ldots, v^1_\beta) \cdots (a^L, v^L_1, \ldots, v^L_\beta) \in ([1, \alpha] \times \mathbb{N}^\beta)^+\)

▶ \(\omega_0 = (4, 1)(3, 2)(4, 2) \in ([1, 4] \times \mathbb{N})^+\).

▶ \(4(=0 + 4) \ 6(=5 + 1) \ 3(=0 + 3) \ 7(=5 + 2) \ 4(=0 + 4) \ 7(=5 + 2).\)
Data words with multiple attributes

\[ (a^1, v^1_1, \ldots, v^1_{\beta}) \cdots (a^L, v^L_1, \ldots, v^L_{\beta}) \in ([1, \alpha] \times \mathbb{N}^\beta)^+ \]

\[ dw_0 = (4, 1)(3, 2)(4, 2) \in ([1, 4] \times \mathbb{N})^+. \]

\[ 4(= 0 + 4) \ 6(= 5 + 1) \ 3(= 0 + 3) \ 7(= 5 + 2) \ 4(= 0 + 4) \ 7(= 5 + 2). \]
Principle of the encoding

- $\omega = (a^1, v^1_1, \ldots, v^1_\beta) \cdots (a^L, v^L_1, \ldots, v^L_\beta) \in [1, \alpha] \times \mathbb{N}^\beta)^+.$

- $\omega$ encoded by $h_{\omega}$ containing a (main) path

$$l^1_0 \rightarrow l^1_1 \rightarrow \cdots \rightarrow l^1_\beta \rightarrow \cdots \rightarrow l^L_0 \rightarrow l^L_1 \rightarrow \cdots \rightarrow l^L_\beta$$

where

- $l^i_0$ has $a^i + 2$ predecessors, for every $i \in [1, L]$.

- $l^i_j$ has $v^i_j + \alpha + 3$ predecessors, for all $i \in [1, L]$ and all $j \in [1, \beta]$.

- every location in the heap domain is either on that path or points to a location on that path.
Heap for the data word $\omega_0 = (2, 1)(1, 2)(2, 2)$

$(\alpha = 2 \text{ and } \beta = 1)$
Fishbone heaps

- Fishbone heap:
  - (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

▶ Fishbone heap:
  ▶ (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.

\[
\varphi_{fb} \overset{\text{def}}{=} \neg \text{emp} \land (\exists u \neg \text{alloc}(u) \land (\forall \bar{u} \text{ alloc}(\bar{u}) \Rightarrow \text{reach}(\bar{u}, u))) \land \\
\neg (\exists u (\#u^2 \geq 0) \ast (\#u^2 \geq 0))
\]

▶ first\( (u) \overset{\text{def}}{=} (\#u \geq 1) \land \neg (\#u^{-1} \geq 1) \).

▶ last\( (u) \overset{\text{def}}{=} (\#u \geq 1) \land \neg \text{alloc}(u) \).
Reminder 1/2

- $\sim \in \{\leq, \geq, =\}$ and $i, \alpha \geq 0$.

  \[
  \#u^0 \sim \alpha \overset{\text{def}}{=} \#u \sim \alpha \\
  \#u^{i+1} \sim \alpha \overset{\text{def}}{=} \exists \, \overline{u} \, u \leftarrow \overline{u} \land \#u^i \sim \alpha \\
  \#u^{-i-1} \sim \alpha \overset{\text{def}}{=} \exists \, \overline{u} \,(\overline{u} \leftarrow u) \land \#u^{-i} \sim \alpha
  \]

- $\#u^6 \geq 2$ states that there is a (necessarily unique) location at distance 6 from $u$ and its number of predecessors is greater than or equal to 2.

- $\#u^{-5} \leq 2$ states that there is a (not necessarily uniquely) location at distance $-5$ from $u$ and its number of predecessors is not strictly greater than 2.
Reminder 2/2

- $u \neq \overline{u}$ and path from $u$ to $\overline{u}$ and nothing else except loops:

\[
\text{reach}'(u, \overline{u}) \overset{\text{def}}{=} \#u = 0 \land \text{alloc}(u) \land \neg \text{alloc}(\overline{u}) \land \\
\forall \overline{u} ((\text{alloc}(\overline{u}) \land \#\overline{u} = 0) \Rightarrow \overline{u} = u) \land \\
\forall u ((\#u \neq 0 \land u \neq \overline{u}) \Rightarrow (\#u = 1 \land \text{alloc}(u)))
\]
Reminder 2/2

- $u \neq \overline{u}$ and path from $u$ to $\overline{u}$ and nothing else except loops:

$$\text{reach}'(u, \overline{u}) \overset{\text{def}}{=} \#u = 0 \land \text{alloc}(u) \land \neg \text{alloc}(\overline{u}) \land \forall \overline{u} ((\text{alloc}(\overline{u}) \land \#\overline{u} = 0) \Rightarrow \overline{u} = u) \land \forall u ((\#u \neq 0 \land u \neq \overline{u}) \Rightarrow (\#u = 1 \land \text{alloc}(u)))$$

- There is a path from $u$ to $\overline{u}$:

$$\text{reach}(u, \overline{u}) \overset{\text{def}}{=} u = \overline{u} \lor (\top \ast \text{reach}'(u, \overline{u}))$$

- Path from $u$ to $\overline{u}$ and nothing else:

$$\text{sreach}(u, \overline{u}) \overset{\text{def}}{=} \text{reach}(u, \overline{u}) \land \neg (\neg \text{emp} \ast \text{reach}(u, \overline{u}))$$

- $\text{sreach}$: strict reachability (a.k.a. list segment predicate $\text{ls}$).
(α, β)-fishbone heaps

▶ (C0): Fishbone heap.

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

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

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

\[
\varphi(C1) \overset{\text{def}}{=} \exists u \, \text{first}(u) \land (3 \leq \#u \leq \alpha + 2)
\]

\[
\varphi(C2) \overset{\text{def}}{=} \forall u \, (3 \leq \#u \leq \alpha + 2) \Rightarrow \bigwedge_{i \in [1, \beta]} \#u^+i \geq \alpha + 3
\]

\[
\varphi(C3) \overset{\text{def}}{=} \forall u \, (3 \leq \#u \leq \alpha + 2) \Rightarrow ((\lnot \#u^+(\beta + 1) \geq 0) \lor (3 \leq \#u^+(\beta + 1) \leq \alpha + 2))
\]
Characterisation

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

- \( \models \text{dw}(\alpha, \beta) \) iff \( \mathcal{h} \) is an \((\alpha, \beta)\)-fishbone heap.

- \( \text{dw}(\alpha, \beta) \) belongs to \(1\text{SL2}(*)\).

- One-to-one correspondence between data words in \(([1, \alpha] \times \mathbb{N}^\beta)^+\) and \((\alpha, \beta)\)-fishbone heaps. (modulo heap isomorphisms)

- Other encodings are possible but the current one is well-suited for further developments.
Characterisation

\[ \text{dw}(\alpha, \beta) \overset{\text{def}}{=} \varphi_{fb} \land \varphi(C_1) \land \varphi(C_2) \land \varphi(C_3) \]

\[ \text{h} \models \text{dw}(\alpha, \beta) \text{ iff } \text{h} \text{ is an } (\alpha, \beta)\text{-fishbone heap.} \]

\[ \text{dw}(\alpha, \beta) \text{ belongs to } 1\text{SL2}(\ast). \]

\[ \text{One-to-one correspondence between data words in } ([1, \alpha] \times \mathbb{N}^\beta)^+ \text{ and } (\alpha, \beta)\text{-fishbone heaps.} \]
\[ \text{(modulo heap isomorphisms)} \]

\[ \text{Other encodings are possible but the current one is well-suited for further developments.} \]

\[ \text{Runs of Minsky machines (with two counters) are data words with } \beta = 2. \]
Arithmetical Constraints Encoded in 1SL2
Arithmetical constraints at the heart of expressive completeness

- Today, we show how to express in 1SL2 constraints $\#u \leq \#u$, $\#u \leq \#u + 1$ or $\#u \leq \#u - 1$ in $(\alpha, 2)$-fishbone heaps when the locations are on the main path.

- This can be generalised to any constraint $\#u + k \sim \#u + k'$ for any heap.

- Key step to show expressive completeness.

  [Brochenin & Demri & Lozes, IC 12; Demri & Deters, LICS’14]

  $\tilde{\#}u \overset{\text{def}}{=} \text{card}(\{l \in \mathbb{N} \mid h(l) = f(u)\})$
Quite different from these constraints

- **u** has at least \( \beta \) predecessors:

\[ \#u \geq \beta \overset{\text{def}}{=} \left( \exists u \ (u \rightarrow u) \right) \ast \cdots \ast \left( \exists u \ (u \rightarrow u) \right) \]

- **u** has at most \( \beta \) predecessors:

\[ \#u \leq \beta \overset{\text{def}}{=} \neg (\#u \geq \beta + 1) \]

- **u** has exactly \( \beta \) predecessors:

\[ \#u = \beta \overset{\text{def}}{=} (\#u \geq \beta) \land \neg (\#u \geq \beta + 1) \]
First principles

- $N \geq M$ iff for all $n \geq 0$, $n \geq N$ implies $n \geq M$.

- Equivalences between
  1. $\tilde{u} \leq \tilde{u}$.
  2. for all $n \in \mathbb{N}$, $n \geq \tilde{u}$ implies $n \geq \tilde{u}$.

Universal quantification simulated by $\forall u$. 
First principles

- $N \geq M$ iff for all $n \geq 0$, $n \geq N$ implies $n \geq M$.

- Equivalences between
  1. $\tilde{u} \leq \tilde{\overline{u}}$.
  2. for all $n \in \mathbb{N}$, $n \geq \tilde{\overline{u}}$ implies $n \geq \tilde{u}$.

- Universal quantification simulated by $\rightarrow^\ast$. 
3 knives, 2 forks and 4 endpoints
A few basic properties

- Any \((\alpha, 2)\)-fishbone heap has no knife and no fork.

- Collection of knives: there is no location in \(\text{dom}(h)\) that does not belong to a knife and no distinct knives share the same endpoint.
A few basic properties

- Any \((\alpha, 2)\)-fishbone heap has no knife and no fork.

- Collection of knives: there is no location in \(\text{dom}(\mathcal{H})\) that does not belong to a knife and no distinct knives share the same endpoint.

- Segmented heap: \(\text{dom}(\mathcal{H}) \cap \text{ran}(\mathcal{H}) = \emptyset\) and no location has strictly more than one predecessor.

\[
\text{seg} \overset{\text{def}}{=} \forall \ u \ \overline{u} \ (u \leftrightarrow \overline{u} \Rightarrow ((\# \overline{u} = 1) \land \#u = 0) \land \neg \text{alloc}(\overline{u})).
\]
A few basic properties

- Any \((\alpha, 2)\)-fishbone heap has no knife and no fork.

- Collection of knives: there is no location in \(\text{dom}(h)\) that does not belong to a knife and no distinct knives share the same endpoint.

- Segmented heap: \(\text{dom}(h) \cap \text{ran}(h) = \emptyset\) and no location has strictly more than one predecessor.

\[
\text{seg} \overset{\text{def}}{=} \forall u \overline{u} (u \leftrightarrow \overline{u} \Rightarrow ((\#u = 1) \land (\#u = 0) \land \neg \text{alloc}(\overline{u}))).
\]

- If \(h' \perp h\), \(h\) is a \((\alpha, 2)\)-fishbone heap and \(h'\) is segmented, then \(h \uplus h'\) has no fork.
Statements about collections

- \( h \models \text{forky}(u) \) iff all the predecessors of \( f(u) \) are endpoints of forks.

\[
\text{forky}(u) \overset{\text{def}}{=} \forall \overline{u} (\overline{u} \leftrightarrow u) \Rightarrow (\exists u (u \leftrightarrow \overline{u}) \land (\#u = 2) \land \neg (\#u^{-1} \geq 1))
\]
 Statements about collections

- $\mathcal{I} \models \text{forky}(u)$ iff all the predecessors of $f(u)$ are endpoints of forks.

$$\text{forky}(u) \overset{\text{def}}{=} \forall \overline{u} (\overline{u} \leftrightarrow u) \Rightarrow (\exists u (u \leftrightarrow \overline{u}) \land (\#u = 2) \land \neg (\#u^{-1} \geq 1))$$

- $\mathcal{I} \models \text{KS}$ iff $\mathcal{I}$ is a collection of knives.

- a knife = $l \overset{\text{part 1}}{\longrightarrow} l' \overset{\text{part 2}}{\longrightarrow} l''$.

$$\text{KS} \overset{\text{def}}{=} \forall u \text{alloc}(u) \Rightarrow (\varphi_{\text{part 1}}(u) \lor \varphi_{\text{part 2}}(u))$$

$$\varphi_{\text{part 1}}(u) \overset{\text{def}}{=} (\#u = 0) \land (\#u^{+1} = 1) \land (\#u^{+2} = 1) \land \neg (\#u^{+3} \geq 0)$$

$$\varphi_{\text{part 2}}(u) \overset{\text{def}}{=} (\#u = 1) \land (\#u^{-1} = 0) \land (\#u^{+1} = 1) \land \neg (\#u^{+2} \geq 0).$$
About the formula $KS1F$

$h \models KS1F$ iff there are $h_1, h_2$ such that

- $h = h_1 \cup h_2$,

- $h_1$ is a collection of knives and,

- $h_2$ is made of a unique fork such that its unique endpoint is not in the range of $h_1$.

See the lecture notes for the details.
Adding a segmented heap for checking \( \#u \leq \#\overline{u} \)

If \( \overline{u} \) can be made forky then \( u \) can be made forky too!

1 memory cell + 1 knife may give 1 fork.
A more realistic illustration
A more realistic illustration
A more realistic illustration
A more realistic illustration
Encoding a single comparison

Assumptions:

- $h_1$ is a $(\alpha, 2)$-fishbone.

- $h_2 \models f \text{seg} \land \#u = 0$.

- $h = h_1 \cup h_2$.

- $f(u)$ is on the main path of $h_1$.

- $n = \text{card}(\text{ran}(h_2) \setminus \text{dom}(h_1))$.

- $m$ is the number of predecessors of $f(u)$ in $h_1$.

- $h \models f \text{KS} \stackrel{*}{\rightarrow} \text{forky}(u)$ iff $n \geq m$.

- $h \models f \text{KS1F} \stackrel{*}{\rightarrow} \text{forky}(u)$ iff $n \geq m - 1$.

(one predecessor of $u$ is possibly taken care by a fork)
Arithmetical constraints in 1SL2

\[
\begin{align*}
\chi_1(u, \bar{u}) & \overset{\text{def}}{=} \text{seg} \land \#u = 0 \land \#ar{u} = 0 \\
\chi_2(u) & \overset{\text{def}}{=} \text{KS} \neg \neg \text{forky}(u) \\
\chi_3(u) & \overset{\text{def}}{=} \text{KS1F} \neg \neg \neg \text{forky}(u).
\end{align*}
\]

- $(\alpha, 2)$-fishbone heap $h_1$ with $f(u)$, $f(\bar{u})$ on the main path.

- $h_1 \models_f \chi_1(u, \bar{u}) \rightarrow (\chi_2(\bar{u}) \Rightarrow \chi_2(u))$ iff $\tilde{\#}u \leq \tilde{\#}\bar{u}$.

- $h_1 \models_f \chi_1(u, \bar{u}) \rightarrow (\chi_2(\bar{u}) \Rightarrow \chi_3(u))$ iff $\tilde{\#}u \leq \tilde{\#}\bar{u} + 1$.

- $h_1 \models_f \chi_1(u, \bar{u}) \rightarrow (\chi_3(\bar{u}) \Rightarrow \chi_2(u))$ iff $\tilde{\#}u \leq \tilde{\#}\bar{u} - 1$. 
\( h_1 \models \chi_1(u, \overline{u}) \prec (\chi_2(\overline{u}) \Rightarrow \chi_3(u)) \) iff \( \#u \leq \#\overline{u} + 1 \)

- \( h_1 \models \left( \chi_1(u, \overline{u}) \prec (\chi_2(\overline{u}) \Rightarrow \chi_3(u)) \right) \).

- For every \( h_2 \bot h_1 \) such that \( h_2 \models \chi_1(u, \overline{u}) \), if \( h_1 \uplus h_2 \models \chi_2(\overline{u}) \), then \( h_1 \uplus h_2 \models \chi_3(u) \).

- For \( n \geq 0 \), there is \( h_2 \bot h_1 \) with

\[
\text{card}(\text{ran}(h_2) \setminus \text{dom}(h_1)) = n
\]

such that \( h_2 \models \chi_1(u, \overline{u}) \) and if \( h_1 \uplus h_2 \models \chi_2(\overline{u}) \), then \( h_1 \uplus h_2 \models \chi_3(u) \).

- for all \( n \geq 0 \), we have \( n \geq \#\overline{u} \) in \( h_1 \) implies \( n \geq \#u - 1 \) in \( h_1 \).

- \( \#u - 1 \leq \#\overline{u} \).
Going even further

- Constraints of the form $\#u + k \sim \#u + k'$ can be expressed in $1SL(\neg \ast)$ for any heap. [Brochenin & Demri & Lozes, IC 12]

- This is even possible in the fragment $1SL2(\neg \ast)$. [Demri & Deters, LICS’14]
Encoding Runs of Minsky Machines in 1SL2
Reminder about Minsky machines

- Minsky machine: finite-state automaton with two counters.
  - Operations on counters:
    - Check whether the counter is zero.
    - Increment the counter by one.
    - Decrement the counter by one if nonzero.
  - Configurations are elements of $[1, \alpha] \times \mathbb{N} \times \mathbb{N}$.
  - Initial configuration: $(1, 0, 0)$.
  - Example of computation:
    $$(1, 0, 0) \rightarrow (2, 1, 0) \rightarrow (1, 1, 1) \rightarrow (2, 2, 1) \rightarrow (1, 2, 2) \rightarrow (2, 3, 2) \ldots$$
Halting problem for Minsky machines

- Runs of Minsky machines are \((\alpha, 2)\)-data words.

- \((\alpha, 2)\)-fishbone heaps can be encoded in 1SL2(*).

- Comparing the number of predecessors for locations at distance three can be done in 1SL2.

- Undecidability of 1SL2 by reduction from the halting problem. [Demri & Deters, TOCL 15]
Encoding $(1, 0, 0) \rightarrow (2, 1, 0) \rightarrow (1, 1, 1) \rightarrow (2, 2, 1)$

Sequence of numbers of predecessors with $\alpha = 3$:

$(3, 6, 6)$ $(4, 7, 6)$ $(3, 7, 7)$ $(4, 8, 7)$
\((\alpha, 2)\)-fishbone heap and limit conditions

- The first three locations on the main path have 3, \(\alpha + 3\), and \(\alpha + 3\) predecessors respectively:

\[
\varphi_{\text{first}} \overset{\text{def}}{=} \exists u \text{ first}(u) \land (#u = 3) \land (#u^+1 = \alpha + 3) \land (#u^+2 = \alpha + 3)
\]

- The main path encoding the run ends by a configuration with the halting instruction:

\[
\varphi_{\text{last}} \overset{\text{def}}{=} \exists u ((#u = \alpha + 2) \land (#u^+2 \geq 0) \land \neg(#u^+3 \geq 0))
\]

- \(\varphi^* \overset{\text{def}}{=} \text{dw}(\alpha, 2) \land \varphi_{\text{first}} \land \varphi_{\text{last}}\).

- \(\models \varphi^*\) iff \(\models \) encodes a data word

\[
\text{dw} = (a^1, v^1_1, v^1_2) \cdots (a^L, v^L_1, v^L_2)
\]

such that \(a^1 = 1\), \(a^L = \alpha\), and \(v^1_1 = v^1_2 = 0\).
Encoding each instruction

For every $I \in [1, \alpha - 1]$, we build $\varphi_I$ so that $M$ halts iff

$$
\varphi^* \land \bigwedge_{I \in [1, \alpha - 1]} \chi_I
$$

is satisfiable in $1\text{SL}2$.

For “$I: c_j := c_j + 1; \text{goto} \ J$” the following properties are to be stated:

1. If $l$ on the main path verifies $\tilde{\#}l = l + 2$ and $h^3(l)$ is defined, then $\tilde{\#}h^3(l) = J + 2$.

2. If a location $l$ encodes the value for the counter $c_j$ in a configuration with instruction $I$ and $h^3(l)$ is defined, then $\tilde{\#}l + 1 = \tilde{\#}h^3(l)$.

3. If $l$ encodes the value for the counter $c_{3-j}$ in a configuration with instruction $I$ and $h^3(l)$ is defined, then $\tilde{\#}l = \tilde{\#}h^3(l)$.
The formula $\chi_I$

\[ \forall u \ (\#u^+ \geq 0) \Rightarrow [((\#u = l + 2) \Rightarrow (\#u^+ = j + 2))] \land \\
((\#u \geq \alpha + 3) \land (\#u^-j = l + 2) \Rightarrow (\#u^+ = \#u + 1)) \land \\
((\#u \geq \alpha + 3) \land (\#u^{j-3} = l + 2) \Rightarrow (\#u^+ = \#u))] \]

$\#u^+ + 3 = J + 2$ is easy to express, see Lecture 1.

$\#u^+ + 3 = \#u + 1$ is difficult to express, see today's lecture.
The formula $\chi_I$

\[
\forall u \ (\#u^3 \geq 0) \Rightarrow \left[ \left( (\#u = I + 2) \Rightarrow (\#u^3 = J + 2) \right) \land \left( (\#u \geq \alpha + 3) \land (\#u^{-j} = I + 2) \Rightarrow (\#u^3 = \#u + 1) \right) \land \left( (\#u \geq \alpha + 3) \land (\#u^{j-3} = I + 2) \Rightarrow (\#u^3 = \#u) \right) \right]
\]

- $\#u^3 = J + 2$ is easy to express, see Lecture 1.

- $\#u^3 = \#u + 1$ is difficult to express, see today’s lecture.
Let’s be more precise!

- We know how to express in 1SL2 constraints \( \#u = \#u \) and \( \#u = \#u + 1 \) in \((\alpha, 2)\)-fishbone heaps when the locations are on the main path.

- But \( \#u + 3 = \#u + 1 \) refers to a location and to another location at distance 3 (and we have only two quantified variables).

- That is easy to handle with three quantified variables:

\[
\exists u' (u \rightarrow u' \land \exists \overline{u} (u' \rightarrow \overline{u} \land \exists u' (\overline{u} \rightarrow u' \land \#u = \#u' - 1))).
\]
Principles to express $\#u = \#u^3 + 3 - 1$

$$\models f(u) = f$$

$$\models \neg (\#u^3 \geq 0) \land (\#u \geq \alpha + 3)$$

$$\models f(\#u^4 = 1) \land \neg (\#u^5 \geq 0)$$

$$\models 1 \text{comp}$$

$$\models \neg \#u^4 \geq 0$$

$$\models \text{size} = 1$$

www.lsv.fr/~demri/essllli15-course.html
“\( I: \) if \( c_j = 0 \) then goto \( J_1 \) else \((c_j := c_j - 1; \) goto \( J_2)\)”

\[
\forall u \ (\#u^+^3 \geq 0) \Rightarrow [((\#u = l + 2) \land (\#u^+^j = \alpha + 3) \Rightarrow (\#u^+^3 = J_1 + 2)) \land
\]

\[
\quad [((\#u = l + 2) \land (\#u^+^j > \alpha + 3) \Rightarrow (\#u^+^3 = J_2 + 2)) \land
\]

\[
\quad [((\#u > \alpha + 3) \land (\#u^{-}^j = l + 2) \Rightarrow (\#u^+^3 = \#u - 1)) \land
\]

\[
\quad [((\#u = \alpha + 3) \land (\#u^{-}^j = l + 2) \Rightarrow (\#u^+^3 = \alpha + 3)) \land
\]

\[
\quad [((\#u \geq \alpha + 3) \land (\#u^{-}^j = l + 2) \Rightarrow (\#u^+^3 = \#u))]\]
Undecidability of 1SL2

- $M$ has a halting run iff

$$dw(\alpha, 2) \land \varphi_{\text{first}} \land \varphi_{\text{last}} \land \bigwedge_{l \in [1, \alpha]} \chi_l$$

is satisfiable in 1SL2.

- 1SL2 satisfiability problem is undecidable.

[Demri & Deters, TOCL 15]

- In the proof both $\ast$ and $\ast \ast$ are used.

- Next, we establish expressive completeness for 1SL2($\ast \ast$) that entails undecidability for 1SL2($\ast \ast$) too.
Expressive Completeness of $k$SL
Expressive completeness

- LTL and first-order logic. [Kamp, PhD thesis 68]

- Extended Temporal Logic ETL and $\omega$-regular languages. [Wolper, IC 83]
Expressive completeness

- LTL and first-order logic.  
  [Kamp, PhD thesis 68]

- Extended Temporal Logic ETL and $\omega$-regular languages.  
  [Wolper, IC 83]

- Bisimulation invariant fragment of FO and modal logic.  
  [van Benthem, Book 83]

- Bisimulation invariant fragment of MSO and modal $\mu$-calculus.  
  [Janin & Walukiewicz, CONCUR’96]
Expressive completeness

- LTL and first-order logic.  
  [Kamp, PhD thesis 68]

- Extended Temporal Logic ETL and $\omega$-regular languages.  
  [Wolper, IC 83]

- Bisimulation invariant fragment of FO and modal logic.  
  [van Benthem, Book 83]

- Bisimulation invariant fragment of MSO and modal $\mu$-calculus.  
  [Janin & Walukiewicz, CONCUR’96]

- Boolean modal logic with converse and identity as expressive as FO2.  
  [Lutz & Sattler & Wolter, CSL’01]
Weak second-order logic $k$WSOL

$$\pi ::= e = e' \mid e \leftrightarrow e_1, \ldots, e_k \mid P(e_1, \ldots, e_n) \mid \text{emp} \mid \bot$$

$$\varphi, \psi ::= \pi \mid \varphi \land \psi \mid \neg \varphi \mid \exists u \varphi \mid \exists P \varphi$$

where $P \in \text{SVAR}_n$ for some $n \geq 1$.

- $(s, h) \models \exists P \varphi$ iff there is a finite $R \subseteq \mathbb{N}^n$ such that $(s, h) \models_{[P \mapsto R]} \varphi$.

- $(s, h) \models P(u_1, \ldots, u_n)$ iff $(V(u_1), \ldots, V(u_n)) \in V(P)$.

- $k$DSOL: Dyadic fragment of $k$WSOL.

- Known reduction from $k$WSOL to $k$DSOL.
From 1SL to 1DSOL
(internalisation of 1SL semantics)

\[ h_p(P) \overset{\text{def}}{=} \forall u, u', u'' (P(u, u') \land P(u, u'')) \Rightarrow u' = u'' \]

\[ P = Q \ast R \overset{\text{def}}{=} \forall u, u' (P(u, u') \iff (Q(u, u') \lor R(u, u')) \land \neg(Q(u, u') \land R(u, u'))) \]

\[ \textbf{Translation} \exists P \left( \forall u, u' P(u, u') \iff u \leftrightarrow u' \right) \land \text{tr}_P(\varphi): \]

\[ \text{tr}_P(u \leftrightarrow u') \overset{\text{def}}{=} P(u, u') \]

\[ \text{tr}_P(\psi \ast \chi) \overset{\text{def}}{=} \exists Q, Q' P = Q \ast Q' \land \text{tr}_Q(\psi) \land \text{tr}_{Q'}(\chi) \]

\[ \text{tr}_P(\psi \ast \chi) \overset{\text{def}}{=} \forall Q \left((\exists Q' h_p(Q') \land Q' = Q \ast P) \land h_p(Q) \land \text{tr}_Q(\psi)) \right. \]

\[ \Rightarrow (\exists Q' h_p(Q') \land Q' = Q \ast P \land \text{tr}_{Q'}(\chi)) \]
Internalisation of 1SL semantics into DSOL.

From $k$WSOL to $k$DSOL

- $P(u) \mapsto P^\text{new}(u, u)$.

- $P(u_1, \ldots, u_n) \mapsto \exists u \bigwedge_{i=1}^{n} P^\text{new}_i(u, u_i)$.

So, it remains to show how to encode 1DSOL into $1\text{SL}2(\neg *)$. 
Principles to reduce 1DSOL into 1SL2(\(\neg\ast\))
(mainly those from [Brochenin & Demri & Lozes, IC 12])

- Valuation heap encodes first-order and second-order valuations.
- Pair \((l, l')\) belongs to \(P_i\) whenever \(l\) and \(l'\) can be identified thanks to some special patterns with arithmetical constraints on the number of predecessors.
- To be able to distinguish the original heap from the valuation heap.
- To be able to have distinct patterns for different variables.
\{2, 5, 7, 9\}-well-formed heap
$j$-parentheses of degree 3 and 5 (plus 1 entry of degree 4)
Well-formed heaps

- $X \subseteq [0, K]$ ($[1, K]$: set of variable indices).

- $X$-well-formed heap:
  1. original heap $h_B \uplus$
  2. valuation heap $h_V$ encoding first-order and second-order valuations for variables with index in $X \setminus \{0\}$.

- Valuation heap $h_V$ is made of
  1. entries,
  2. left parentheses or right parentheses.

(respective degrees are greater than any degree in the original heap)
Well-formed heaps

- $X \subseteq [0, K]$ ($[1, K]$: set of variable indices).

- $X$-well-formed heap:
  1. original heap $h_B \uplus$
  2. valuation heap $h_V$ encoding first-order and second-order valuations for variables with index in $X \setminus \{0\}$.

- Valuation heap $h_V$ is made of
  1. entries,
  2. left parentheses or right parentheses.

(respective degrees are greater than any degree in the original heap)

- Valuation heap is a collection of local valuations (one per variable).
Well-formed heaps

- $X \subseteq [0, K]$ ($[1, K]$: set of variable indices).

- $X$-well-formed heap:
  1. original heap $h_B \uplus$
  2. valuation heap $h_V$ encoding first-order and second-order valuations for variables with index in $X \setminus \{0\}$.

- Valuation heap $h_V$ is made of
  1. entries,
  2. left parentheses or right parentheses. (respective degrees are greater than any degree in the original heap)

- Valuation heap is a collection of local valuations (one per variable).

- Separation between $h_B$ and $h_V$ is concretised by the existence of the left parenthesis with index 0 that have the maximal degree among thoses parentheses.
Encoding valuations
Satisfaction of $P_i(u_j, u_k) \ (j < i < k)$
Translation into 1SL2(\(\neg\ast\))

\[
\begin{align*}
t(P_i(u_j, u_k)) & \overset{\text{def}}{=} \exists u \ (\text{on}_j(u) \land \exists \bar{u} \ (\bar{u} \leftrightarrow u) \land \\
& \text{vind}_i(\bar{u}) \land \exists u \ (\#u = \#\bar{u} + 1 \land \text{vind}_i(u) \land \exists \bar{u} \ (u \leftrightarrow \bar{u} \land \text{on}_k(\bar{u}))))
\end{align*}
\]
More about the translation

\[ T(\varphi) \overset{\text{def}}{=} \exists u \text{ isoloc}(u) \land (\text{localval}_0(u) \not\Rightarrow \bigg( \text{wfh}_0 \land \text{imin}(u) \land (\forall \overline{u} ((u \neq \overline{u}) \land \neg \text{lrp}_0(\overline{u})) \Rightarrow (\#\overline{u} < \#u) \land t(\{0\}, \varphi) \bigg) \bigg) \}

well–formedness and maximality
More about the translation

\[
T(\varphi) \overset{\text{def}}{=} \exists u \isoloc(u) \land \left( \text{localval}_0(u) \stackrel{\text{new}}{\Rightarrow} \right)
\\
(\text{wfh}_0 \land \text{imin}(u) \land (\forall \overline{u} ((u \neq \overline{u}) \land \neg \text{lrp}_0(\overline{u})) \Rightarrow (\#\overline{u} < \#u) \land t(\{0\}, \varphi)))
\\
\text{well–formedness and maximality}
\]

\[
t(X, u_i = u_j) \overset{\text{def}}{=} \exists u \on_{i}(u) \land \on_{j}(u)
\\
t(X, u_i \leftrightarrow u_j) \overset{\text{def}}{=} \exists u \exists \overline{u} (\on_{i}(u) \land \on_{j}(\overline{u}) \land u \leftrightarrow \overline{u})
\]
More about the translation

\[
T(\varphi) \overset{\text{def}}{=} \exists u \ \text{isoloc}(u) \land (\text{localval}_0(u) \not\to^{*} \\
(\text{wfh}_{\{0\}} \land \text{imin}(u) \land (\forall \overline{u} \ ((u \neq \overline{u}) \land \neg \text{lrp}_0(\overline{u}))) \Rightarrow (\#\overline{u} < \#u) \land t(\{0\}, \varphi)))
\]

well-formedness and maximality

\[
t(X, u_i = u_j) \overset{\text{def}}{=} \exists u \ \text{on}_{i}(u) \land \text{on}_{j}(u)
\]

\[
t(X, u_i \leftarrow u_j) \overset{\text{def}}{=} \exists u \exists \overline{u} \ (\text{on}_{i}(u) \land \text{on}_{j}(\overline{u}) \land u \leftarrow \overline{u})
\]

\[
t(X, \exists u_i \psi) \overset{\text{def}}{=} \exists u \exists \overline{u} \\
((\text{imin}(u) \land \text{isoloc}(\overline{u})) \land (\text{localval}_i(\overline{u}) \not\to^{*} \\
(\text{wfh}_{X \cup \{i\}} \land \text{imin}(u) \land \text{llp}_i(\overline{u}) \land t(X \cup \{i\}, \psi))))
\]

\[
t(X, \exists P_i \psi) \overset{\text{def}}{=} \exists u \exists \overline{u} \\
((\text{imin}(u) \land \text{isoloc}(\overline{u})) \land (\text{localval}_i(\overline{u}) \not\to^{*} \\
(\text{wfh}_{X \cup \{i\}} \land \text{imin}(u) \land \text{llp}_i(\overline{u}) \land t(X \cup \{i\}, \psi))))
\]
Properties of the translation

- \( \psi \) subformula of \( \varphi \) with \((\text{free}(\psi) \cup \{0\}) \subseteq X \subseteq [0, K]\). 
  \( X \)-well-formed \( h = h_B \cup h_V \) and extracted valuation \( \mathcal{V}_h \). 
  Then, \( h_B \models \mathcal{V}_h \psi \) iff \( h \models t(X, \psi) \).

- For every sentence \( \varphi \) in DSOL, for every heap \( h \), we have 
  \( h \models \varphi \) iff \( h \models \mathcal{T}(\varphi) \).

- Detailed developments can be found in the lecture notes.
A translation $\top$

- For every sentence $\varphi$ in DSOL, for every heap $h$, we have $h \models \varphi$ iff $h \models \top(\varphi)$.

- 1WSOL and 1SL2($\neg\ast$) have the same expressive power.

  [Demri & Deters, CSL-LICS’14]

- ... but 1SL($\ast$) is decidable by translation into weak monadic second-order theory of $(D, f, =)$.

  [Rabin, Trans. of AMS 69]

See Lecture 4
For every sentence $\varphi$ in DSOL, for every heap $h$, we have $h \models \varphi$ iff $h \models T(\varphi)$.

1WSOL and 1SL2($\neg\ast$) have the same expressive power. \[\text{[Demri & Deters, CSL-LICS’14]}\]

Satisfiability problem for 1SL2($\neg\ast$) is undecidable.

Validity in 1SL2($\neg\ast$) is not recursively enumerable.

...but 1SL($\ast$) is decidable by translation into weak monadic second-order theory of $(D, f, =)$.

[Rabin, Trans. of AMS 69]

See Lecture 4
No proof systems for $1SL_2(\rightarrow^*)$ on concrete heaps!

- $1SL$ is not finitely axiomatizable.
- This applies to $2SL$ too!
No proof systems for $1SL_2(\rightarrow \star)$ on concrete heaps!

- 1SL is not finitely axiomatizable.

- This applies to 2SL too!

- First-order theory of $(\mathbb{N}, \leq, \times, +)$ is not recursively enumerable by Gödel’s first incompleteness theorem.

- Reduction from first-order arithmetics to 1WSOL.

- Reduction from 1WSOL to 1SL.
Conclusion

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

- Tomorrow’s lecture:
  - Undecidability proofs by reduction from data logics.
  - Non-elementarity proof by reduction from propositional interval temporal logic.
  - A modal logic of heap.
Conclusion

▶ Today’s lecture:
  ▶ Data words and heaps.
  ▶ Arithmetical constraints encoded in 1SL2.
  ▶ Undecidability of 1SL2.
  ▶ Expressive completeness of 1SL.

▶ Tomorrow’s lecture:
  ▶ Undecidability proofs by reduction from data logics.
  ▶ Non-elementarity proof by reduction from propositional interval temporal logic.
  ▶ A modal logic of heap.