Confluence: decidability results & what to do without

Florent Jacquemard

5th International Workshop on Confluence - IWC’16
1. (un)decidability of confluence
   - map of results on (un)decidability of confluence and Uniqueness of Normal Forms
   - importance of linearity
   - open problems
   - links to regularity preservation

2. enumeration of equivalent terms
   - case study: terms-based representation of durations in music western notation
   - non-confluent TRS
   - automata-based representation of equivalent terms sets
   - lazy ordered enumeration of equivalence classes
   - links to regularity preservation
undecidability

decidability

confluence

ground TRS
[Oyamaguchi 87]
[Dauchet Tison 88]

left-linear
right-ground TRS
[Dauchet et al 90]

linear TRS (depth 2)
[Verma et al 01]

flat TRS
[03, 06, 09]

EXPTIME

PTIME

ground TRS
[Comon et al 01]
[Tiwari 02]

left-linear
right-ground TRS
[Dauchet et al 90]

length 2 SRS
[Sakai 07]

ground TRS
[Comon et al 01]
[Tiwari 02]

left-shallow-linear
right-ground TRS
[Tiwari 02]

linear-shallow TRS
[Tiwari 02]
[Godoy et al 04]

every variable occurs at most once in each rule and at depth at most 1

every variable occurs at most once in each rule and at depth at most 1 in each side of rule

linear and shallow TRS
[Godoy et al 03]
Undecidability

Decidability

Confluence

Ground TRS
[Oyamaguchi 87] [Dauchet Tison 88]

Left-linear right-ground TRS
[Dauchet et al 90]

Linear TRS (depth 2)
[Verma et al 01]

Linear and shallow TRS
[Godoy et al 03]

Linear-shallow-linear right-ground TRS
[Tiwari 02]

Left-shallow-linear right-ground TRS
[Tiwari 02]

Linear-shallow TRS
[Tiwari 02]

Ground TRS
[Comon et al 01] [Tiwari 02]

PTIME

EXPTIME

Flat TRS
[03, 06, 09]

Length 2 SRS
[Sakai 07]

Every variable occurs at most once in each rule
and at depth at most 1

Every variable occurs at most once in each side of
rule

Regular closure preserves:
$\text{closure}_R(L)$ is regular when $L$ is regular
undecidability of confluence for flat TRS

[Jacquemard 03, Mitsuhashi, Oyamaguchi, J. 06 ]

1. undecidability of reachability:
   PCP encoding (*shifted pairing* technique)
2. reduction of reachability to confluence

\[ \begin{array}{c}
  \text{u}_1 \\
  \vdots \\
  \text{u}_m \\
\end{array} \quad \overrightarrow{\Pi_1} \quad \begin{array}{cc}
  \text{u}_1 & \text{V}_1 \\
  \vdots & \vdots \\
  \text{u}_m & \text{V}_m \\
  \_ & \text{V}_{m+1} \\
  \_ & \_ \\
  \_ & \text{V}_n \\
\end{array} \quad \overrightarrow{\Pi_2} \quad \begin{array}{c}
  \text{V}_1 \\
  \vdots \\
  \text{V}_m \\
  \_ \\
  \_ \\
  \_ \\
  \_ \\
  \text{V}_n \\
\end{array} \]

simpler proofs in [Godoy Hernandez 09]
undecidability of confluence for flat TRS

[Jacquemard 03, Mitsuhashi, Oyamaguchi, J. 06 ]

1. undecidability of reachability:
   PCP encoding (*shifted pairing* technique)
2. reduction of reachability to confluence

\[
0 \rightarrow f( q_A^{(3)}, q_A^{(4)}, q_A^{(5)}, q_B^{(13)}, q_B^{(14)}, q_A^{(15)}, q_B^{(16)} )
\]
\[
T_A^{(3)} T_A^{(4)} T_A^{(5)} T_B^{(13)} T_B^{(14)} T_A^{(6)} T_B^{(15)} T_B^{(16)}
\]
\[
P^{(3,1)} P^{(4,2)} P^{(5,1)} S^{(13,11)} S^{(14,12)} P^{(6,2)} S^{(15,11)} S^{(16,12)}
\]
\[
f( x_1, x_2, x_1, y_{11}, y_{12}, x_2, y_{11}, y_{12} )
\]
\[
g( x_1, x_2, x_1, y_{11}, y_{12}, x_2, y_{11}, y_{12} )
\]
\[
P^{(1,0)} P^{(2,0)} \Pi_1^{(1,17)} S^{(11,17)} S^{(12,18)} \Pi_2^{(2,18)} S^{(11,10)} S^{(12,10)}
\]
\[
g( x_0, x_0, y_{17}, y_{17}, y_{18}, y_{18}, y_{10}, y_{10} ) \rightarrow 1
\]

simpler proofs in [Godoy Hernandez 09]
decidability

**confluence**
(non-linear TRSs)

right-ground TRS
[Kaiser 05]

shallow and right-linear TRS
[Godoy, Tiwari 05]

flat TRS
[03, 06, 09]

undecidability
uniqueness of NF

uniqueness of NF (UN=): no two distinct normal forms can be equivalent modulo the TRS.

confluence $\Rightarrow$ UN=

unique normalization (UN): every term can reach at most one normal form using the TRS.

UN= $\Rightarrow$ UN (the converse is not true)
undecidability

ground TRS [Verma 08]

shallow and linear TRS [Verma, Zinn 06]

shallow TRS [Radcliffe, Verma 10]

ground TRS [Verma, Hayrapetyan 05]

shallow and linear TRS [Godoy, J. 09]

right-ground (right-flat) TRS [Verma 08,09]

linear, non-collapsing, var-preserving, depth 2 TRS [Verma 08]

linear, left-flat, right-depth 2 TRS [Radcliffe, Verma 10]

right-ground TRS [Verma 08]

flat TRS [Godoy, Hernández 09]

linear and right-flat TRS [Godoy, Tison 07]

flat and right-linear TRS [Godoy, J. 09]
confluence and UN under rewrite strategies

[Durand, Sénizergues 07]

new

[Ishizuki, Sakai, Oyamaguchi IWC 16]

conditions for confluence of innermost-terminating TRS

for bottom-up term rewriting?

[Durand, Sénizergues 07]
open questions: decidability of confluence for flat and non-collapsing TRS?

\[
\Pi_1 := \left\{ \langle a, b \rangle(x) \to a(x) \mid a \in \Sigma, b \in \Sigma \cup \{\_\} \right\}
\cup \left\{ \langle \_, b \rangle(x) \to x \mid b \in \Sigma \right\}
\]
\[
\Pi_2 := \left\{ \langle a, b \rangle(x) \to b(x) \mid a \in \Sigma \cup \{\_\} b \in \Sigma \right\}
\cup \left\{ \langle a, \_ \rangle(x) \to x \mid a \in \Sigma \right\}
\]

(collapsing rules in PCP reduction by shifted pairing)

regularity preserving TRSs
- regularity preservation used in decision of confluence, e.g.
  - local confluence:
    \[
    s_1 \xleftarrow{\mathcal{R}} t \xrightarrow{\mathcal{R}} s_2 \Rightarrow \{ s'_1 \mid s_1 \xrightarrow{\mathcal{R}} s'_1 \} \cap \{ s'_2 \mid s_2 \xrightarrow{\mathcal{R}} s'_2 \} \neq \emptyset
    \]
  - original decidability proofs for ground TRS
- decidability for other regularity preserving TRSs?
  - right-linear and finite-path-overlapping TRS [Takai et al 00]
  - layered traducing TRS [Seki et al 02]
1. (un)decidability of confluence
   • map of results on (un)decidability of confluence and Uniqueness of Normal Forms
   • importance of linearity and flatness
   • open problems
   • links to regularity preservation

2. enumeration of equivalent terms
   • case study: terms-based representation of durations in music western notation
   • non-confluent TRS
   • automata-based representation of equivalent terms sets
   • lazy ordered enumeration of equivalence classes
   • links to regularity preservation
notation of durations in music

In common western music notation, durations are defined relative to a periodic pulse (beat) and hierarchically, by recursive subdivisions (nested fractions).

Rizo
Symbolic music comparison with tree data structures
PhD thesis U. Alicante, 2010
rhythms as syntax trees

Longuet-Higgins
The perception of music
I.S.R., 1978

\[ c \rightarrow \circ | - | \frac{2}{4} + \frac{2}{4} \]
\[ \frac{2}{4} \rightarrow \underline{\text{d}} | - | \frac{1}{4} + \frac{1}{4} \]
\[ \frac{1}{4} \rightarrow \underline{\text{q}} | 8 | \frac{1}{8} + \frac{1}{8} \]
\[ \frac{1}{8} \rightarrow \underline{\text{q}} | 7 | \ldots \]

Lee
The rhythmic interpretation of simple musical sequences
Musical Structure and Cognition, 1985

\[ \text{music notation} \]

▶ symbolic constraints (e.g. sum = 1)
▶ definition of schemas for rhythm notations as regular tree languages
▶ definition of syntactic transformations or equations
rhythm trees (RT): syntax
(simplified version)

hierarchical encoding of durations
as terms over a finite (and small) signature $\Sigma$

• one symbol $p$ of arity $p$ for each $1 < p \leq$ bound (typically 13)
• constant symbols: $n$ (note), $r$ (rest), $o$ ($tie$=composition)
rhythm trees (RT) : semantics

we associate a duration to every node:

\[
dur(root) = 1 \text{ beat or 1 measure}
\]

\[
dur(node) = \frac{dur(parent)}{arity(parent)} + pdur(node)
\]

\[
pdur(node) = \begin{cases} 
\text{dur(next-leaf)} & \text{if next-leaf exists and is labeled with o} \\
0 & \text{otherwise}
\end{cases}
\]

rhythmic value = sequence of rational numbers
= duration of leaves (in dfs traversal) labelled n or r
we associate a **duration** to every node:

\[
\text{dur}(\text{root}) = 1 \text{ beat or 1 measure}
\]

\[
\text{dur}(\text{node}) = \frac{\text{dur}(\text{parent})}{\text{arity}(\text{parent})} + \text{pdur}(\text{node})
\]

\[
\text{pdur}(\text{node}) = \text{dur}(\text{next-leaf})
\quad \text{if } \text{next-leaf} \text{ exists and is labeled with } n
\]

\[
\text{pdur}(\text{node}) = 0 \text{ otherwise}
\]

**rhythmic value** = sequence of rational numbers
= duration of leaves (in dfs traversal) labelled **n** or **r**
rhythmic value

rests

notes

\[
\begin{bmatrix}
\frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6}
\end{bmatrix}
\]
rhythmic value (nested tuplets)

\[
\begin{array}{c}
 5 \\
 3 \\
 n \\
\end{array}
\]

rhythmic value

\[
\frac{1}{5} \quad \frac{1}{5} \quad \frac{1}{15} \quad \frac{1}{15} \quad \frac{1}{15} \quad \frac{5}{15} \quad \frac{5}{15}
\]

\[
\begin{array}{c}
 5 \\
 3 \\
\end{array}
\]
ties and dots

we sum durations for subsequences of leafs of the form \( n o \ldots o \)
rewrite rules

Structural Theory of Rhythm Notation MEI’15, MCM’15

addition of rests

\[ p(r, \ldots, r) \rightarrow r \]
\[ p(r, o, \ldots, o) \rightarrow r \]

normalization of ties

\[ p(o, \ldots, o) \rightarrow o \]
\[ p(n, o, \ldots, o) \rightarrow n \]

arity switch

\[ \begin{array}{c}
p \\
q \\
\ldots \\
q
\end{array} \quad \rightarrow \quad \begin{array}{c}
q \\
\ldots \\
p
\end{array} \]

\[ \begin{array}{c}
x_1 \\
\ldots \\
x_q \\
\ldots \\
x_{pq}
\end{array} \quad \rightarrow \quad \begin{array}{c}
x_1 \\
\ldots \\
x_p \\
\ldots \\
x_{pq}
\end{array} \]
peak example
peak example

normalise ties

\[ \begin{array}{c}
\text{normalise ties} \\
\text{normalise ties}
\end{array} \]
peak example
representation and enumeration of equivalence classes

given:
1. a finite description of a set $L$ of allowed RT as CF grammar (RT schema)
2. a RT $t$

return:
• a finite description of the set $L'$ of RT in $L$ of same rhythmic value as $t$

variant:
• $L$ associates to every RT a weight in an ordered semiring
• lazy enumeration of $L'$ according to weight
RT schemas

acyclic CF grammars defining allowed divisions
defined RTs = derivation trees without n.t.
= tree automata language

ex: division by 2 or 3, then 2 then 2

\[
\begin{align*}
q_0 & ::= q_1 q_1 & q_1 & ::= q_2 q_2 & q_2 & ::= q_3 q_3 \\
q_0 & ::= q_1 q_1 q_1 & q_1 & ::= n | r | o & q_2 & ::= n | r | o & q_3 & ::= n | r | o
\end{align*}
\]
construction of schema for equivalence class

1. schema

\[ q_0 ::= q_1 q_1 \quad q_1 ::= q_2 q_2 \quad q_2 ::= q_3 q_3 \]
\[ q_0 ::= q_1 q_1 q_1 \quad q_1 ::= n \mid r \mid o \quad q_2 ::= n \mid r \mid o \quad q_3 ::= n \mid r \mid o \]

2. initial RT of rhythm value \( \frac{1}{4} \frac{1}{4} \frac{1}{2} \)

- target schema for equivalence class

\[
\begin{align*}
[\frac{1}{4} \frac{1}{4} \frac{1}{2}]^{q_0} & ::= \left[\frac{1}{4} \frac{1}{4}\right] q_1 \left[\frac{1}{2}\right] q_1 & \left[\frac{1}{2}\right] q_1 & ::= n \mid r \\
[\frac{1}{4} \frac{1}{4}]^{q_1} & ::= \left[\frac{1}{4}\right] q_2 \left[\frac{1}{4}\right] q_2 & \left[\frac{1}{4}\right] q_2 & ::= n \mid r \\
[\frac{1}{4} \frac{1}{4} \frac{1}{2}]^{q_0} & ::= \left[\frac{1}{4} \frac{1}{12}\right] q_1 \left[\frac{1}{6} \frac{1}{6}\right] q_1 \left[\frac{1}{3}\right] q_1 & \left[\frac{1}{3}\right] q_1 & ::= o \\
[\frac{1}{4} \frac{1}{12}]^{q_1} & ::= \left[\frac{1}{6}\right] q_2 \left[\frac{1}{12} \frac{1}{12}\right] q_2 & \left[\frac{1}{6}\right] q_2 & ::= n \mid r \\
[\frac{1}{12} \frac{1}{12}]^{q_2} & ::= \left[\frac{1}{12}\right] q_3 \left[\frac{1}{12}\right] q_3 & \left[\frac{1}{12}\right] q_3 & ::= o \\
[\frac{1}{12}]^{q_3} & ::= [\frac{1}{12}] q_3 & \left[\frac{1}{12}\right] q_3 & ::= n \mid r \\
[\frac{1}{6}, \frac{1}{6}]^{q_1} & ::= [\frac{1}{6}] q_2 [\frac{1}{6}] q_2 & [\frac{1}{6}] q_2 & ::= o \\
[\frac{1}{6}]^{q_2} & ::= [\frac{1}{6}] q_2 & [\frac{1}{6}] q_2 & ::= n \mid r \\
[\frac{1}{6}]^{q_3} & ::= [\frac{1}{6}] q_3 & [\frac{1}{6}] q_2 & ::= n \mid r
\end{align*}
\]
schema for equivalence class and derivation trees

\[
\begin{align*}
[\frac{1}{4} \frac{1}{4} \frac{1}{2}]_{q_0} & := \quad [\frac{1}{4} \frac{1}{4}]_{q_1} [\frac{1}{2}]_{q_1} \\
[\frac{1}{4} \frac{1}{4}]_{q_1} & := \quad [\frac{1}{4}]_{q_2} [\frac{1}{4}]_{q_2}
\end{align*}
\]
weights and lazy enumeration

• add weights to CFG production rules
  → defines a notion of complexity of RT (size, penalty for tuples…)
• size of derivation tree = product of weights or rules
• size of RT = sum of sizes of weights of matching derivation trees

• lazy enumeration of $k$ best derivation trees
  $k$-best parsing (dynamic programming) [Huang, Chiang 05]
  table based on the target schema (1 row for each NT).

$O(|\text{target schema}| + c_{\text{max}} \cdot k \cdot \log(k))$

$c_{\text{max}}$: max number of production rules for one NT
rhythm dags (RD)

- no symbol \( \bullet \)
- sum of durations represented by node sharing
  (the data is in the structure)
- captures ratio notation (\( p \) in the time of \( q \))

\[
\begin{array}{cccccc}
2 & 2 & r & 3 & r & 2 \\
1 & 1 & 1 & 1 & 1 & 1 \\
\frac{4}{4} & \frac{6}{6} & \frac{6}{6} & \frac{6}{6} & \frac{4}{4} \\
\end{array}
\quad\quad\quad\quad
\begin{array}{cccccc}
2 & 2 & 4 & r & 3 & 3 \\
3 & 3 & 3 & 3 & 3 & 1 \\
\frac{3}{16} & \frac{3}{16} & \frac{3}{16} & \frac{3}{16} & \frac{4}{4} \\
\end{array}
\]
RD schemas

acyclic CS grammars define lang. of dag automata of [Kamimura, Slutzki 1981]

\[
\begin{align*}
\varepsilon & ::= q_1q_1 \\
q_1 & ::= q_2q_2 \\
q_2q_2 & ::= q_3q_3q_3 \quad (3:2)
\end{align*}
\]
OpenMusic Rhythm Trees

**OpenMusic**: graphical programming environment for algorithmic composition developed at Ircam

OM RT (nested lists) are a first class data structure for the representation of rhythms in OM

a library for rhythm transcription [Ycart et al ICMC’16]
Rhythm Trees Applications

- algorithmic composition, music score editors
  transcription, assistance, transformations…
- computational musicology
  analysis of score corpora, data mining
- digital music score databases
  information retrieval, indexing, *query by tapping*
- metrical phonology in speech production
automata-based representations of rewrite closure

(regularity preservation)

given:
• a tree automaton (TA) \( A \) recognizing a set of terms \( L \)
• a TRS \( R \)

return:
• a tree automaton \( A' \) recognizing the forward closure of \( L \) by \( R \)

Used as theoretical tool in some proofs of decidability of confluence

Used as practical tool for enumeration of reachable terms
- e.g. counter examples, error configurations in verification…
Thank You