

Trace monoid

In computer science, a **trace** is a set of strings, wherein certain letters in the string are allowed to commute, but others are not. It generalizes the concept of a string, by not forcing the letters to always be in a fixed order, but allowing certain reshufflings to take place. Traces were introduced by Pierre Cartier and Dominique Foata in 1969 to give a combinatorial proof of MacMahon's Master theorem. Traces are used in theories of concurrent computation, where commuting letters stand for portions of a job that can execute independently of one another, while non-commuting letters stand for locks, synchronization points or thread joins.^[1]

The **trace monoid** or **free partially commutative monoid** is a monoid of traces. In a nutshell, it is constructed as follows: sets of commuting letters are given by an independency relation. These induce an equivalence relation of equivalent strings; the elements of the equivalence classes are the traces. The equivalence relation then partitions up the free monoid (the set of all strings of finite length) into a set of equivalence classes; the result is still a monoid; it is a quotient monoid and is called the *trace monoid*. The trace monoid is universal, in that all dependency-homomorphic (see below) monoids are in fact isomorphic.

Trace monoids are commonly used to model concurrent computation, forming the foundation for process calculi. They are the object of study in trace theory. The utility of trace monoids comes from the fact that they are isomorphic to the monoid of dependency graphs; thus allowing algebraic techniques to be applied to graphs, and vice versa. They are also isomorphic to history monoids, which model the history of computation of individual processes in the context of all scheduled processes on one or more computers.

Contents

- Trace
- Examples
- Properties
- Universal property
- Normal forms
- Trace languages
- Notes
- References

Trace

Let Σ^* denote the free monoid, that is, the set of all strings written in the alphabet Σ . Here, the asterisk denotes, as usual, the Kleene star. An independency relation I on Σ then induces a binary relation \sim on Σ^* , where $u \sim v$ if and only if there exist $x, y \in \Sigma^*$, and a pair $(a, b) \in I$ such that $u = xaby$ and $v = xbay$. Here, u, v, x and y are understood to be strings (elements of Σ^*), while a and b are letters (elements of Σ).

The **trace** is defined as the symmetric, reflexive and transitive closure of \sim . The trace is thus an equivalence relation on Σ^* , and is denoted by \equiv_D . The subscript D on the equivalence simply denotes that the equivalence is obtained from the independency I induced by the dependency D . Clearly, different dependencies will give different equivalence relations.

The transitive closure simply implies that $u \equiv v$ if and only if there exists a sequence of strings (w_0, w_1, \dots, w_n) such that $u \sim w_0$ and $v \sim w_n$ and $w_i \sim w_{i+1}$ for all $0 \leq i < n$. The trace is stable under the monoid operation on Σ^* (concatenation) and is therefore a congruence on Σ^* .

The trace monoid, commonly denoted as $\mathbb{M}(D)$, is defined as the quotient monoid

$$\mathbb{M}(D) = \Sigma^* / \equiv_D.$$

The homomorphism

$$\phi_D : \Sigma^* \rightarrow \mathbb{M}(D)$$

is commonly referred to as the natural homomorphism or **canonical homomorphism**. That the terms *natural* or *canonical* are deserved follows from the fact that this morphism embodies a universal property, as discussed in a later section.

Examples

Consider the alphabet $\Sigma = \{a, b, c\}$. A possible dependency relation is

$$\begin{aligned} D &= \{a, b\} \times \{a, b\} \cup \{a, c\} \times \{a, c\} \\ &= \{a, b\}^2 \cup \{a, c\}^2 \\ &= \{(a, b), (b, a), (a, c), (c, a), (a, a), (b, b), (c, c)\} \end{aligned}$$

The corresponding independency is

$$I_D = \{(b, c), (c, b)\}$$

Therefore, the letters b, c commute. Thus, for example, a trace equivalence class for the string $abababbca$ would be

$$[abababbca]_D = \{abababbca, abababcba, ababacbba\}$$

The equivalence class $[abababbca]_D$ is an element of the trace monoid.

Properties

The **cancellation property** states that equivalence is maintained under right cancellation. That is, if $w \equiv v$, then $(w \div a) \equiv (v \div a)$. Here, the notation $w \div a$ denotes right cancellation, the removal of the first occurrence of the letter a from the string w , starting from the right-hand side. Equivalence is also maintained by left-cancellation. Several corollaries follow:

- **Embedding:** $w \equiv v$ if and only if $xwy \equiv xvy$ for strings x and y . Thus, the trace monoid is a syntactic monoid.

- Independence: if $ua \equiv vb$ and $a \neq b$, then a is independent of b . That is, $(a, b) \in I_D$. Furthermore, there exists a string w such that $u \equiv wb$ and $v \equiv wa$.
- Projection rule: equivalence is maintained under string projection, so that if $w \equiv v$, then $\pi_\Sigma(w) \equiv \pi_\Sigma(v)$.

A strong form of Levi's lemma holds for traces. Specifically, if $uv \equiv xy$ for strings u, v, x, y , then there exist strings z_1, z_2, z_3 and z_4 such that $(w_2, w_3) \in I_D$ for all letters $w_2 \in \Sigma$ and $w_3 \in \Sigma$ such that w_2 occurs in z_2 and w_3 occurs in z_3 , and

$$\begin{aligned} u &\equiv z_1 z_2, & v &\equiv z_3 z_4, \\ x &\equiv z_1 z_3, & y &\equiv z_2 z_4. \end{aligned}^{[2]}$$

Universal property

A **dependency morphism** (with respect to a dependency D) is a morphism

$$\psi : \Sigma^* \rightarrow M$$

to some monoid M , such that the "usual" trace properties hold, namely:

1. $\psi(w) = \psi(\varepsilon)$ implies that $w = \varepsilon$
2. $(a, b) \in I_D$ implies that $\psi(ab) = \psi(ba)$
3. $\psi(ua) = \psi(v)$ implies that $\psi(u) = \psi(v \div a)$
4. $\psi(ua) = \psi(vb)$ and $a \neq b$ imply that $(a, b) \in I_D$

Dependency morphisms are universal, in the sense that for a given, fixed dependency D , if $\psi : \Sigma^* \rightarrow M$ is a dependency morphism to a monoid M , then M is isomorphic to the trace monoid $\mathbb{M}(D)$. In particular, the natural homomorphism is a dependency morphism.

Normal forms

There are two well-known normal forms for words in trace monoids. One is the lexicographic normal form, due to Anatolij V. Anisimov and Donald Knuth, and the other is the *Foata* normal form due to Pierre Cartier and Dominique Foata who studied the trace monoid for its combinatorics in the 1960s.

Trace languages

Just as a formal language can be regarded as a subset of Σ^* the set of all possible strings, so then a trace language is defined as subset of $\mathbb{M}(D)$ all possible traces.

A language $L \subseteq \Sigma^*$ is a trace language, or is said to be **consistent** with dependency D if

$$L = \bigcup [L]_D$$

where

$$[L]_D = \{[w]_D \mid w \in L\}$$

is the trace closure of a set of strings.

Notes

1. Sándor & Crstici (2004) p.161
2. Proposition 2.2, Diekert and Métivier 1997.

References

General references

- Diekert, Volker; Métivier, Yves (1997), "Partial Commutation and Traces" (<http://citeseer.ist.psu.edu/diekert97partial.html>), in Rozenberg, G.; Salomaa, A. (eds.), *Handbook of Formal Languages Vol. 3; Beyond Words*, Springer-Verlag, Berlin, pp. 457–534, ISBN 3-540-60649-1
- Lothaire, M. (2011), *Algebraic combinatorics on words*, Encyclopedia of Mathematics and Its Applications, **90**, With preface by Jean Berstel and Dominique Perrin (Reprint of the 2002 hardback ed.), Cambridge University Press, ISBN 978-0-521-18071-9, Zbl 1221.68183 (<https://zbmath.org/?format=complete&q=an:1221.68183>)
- Antoni Mazurkiewicz, "Introduction to Trace Theory", pp 3–41, in *The Book of Traces*, V. Diekert, G. Rozenberg, eds. (1995) World Scientific, Singapore ISBN 981-02-2058-8
- Volker Diekert, *Combinatorics on traces*, LNCS 454, Springer, 1990, ISBN 3-540-53031-2, pp. 9–29
- Sándor, Jozsef; Crstici, Borislav (2004), *Handbook of number theory II*, Dordrecht: Kluwer Academic, pp. 32–36, ISBN 1-4020-2546-7, Zbl 1079.11001 (<https://zbmath.org/?format=complete&q=an:1079.11001>)

Seminal publications

- Pierre Cartier and Dominique Foata, *Problèmes combinatoires de commutation et réarrangements*, Lecture Notes in Mathematics 85, Springer-Verlag, Berlin, 1969, Free 2006 reprint (<http://www.emis.de/journals/SLC/books/cartfoa.html>) with new appendixes
- Antoni Mazurkiewicz, *Concurrent program schemes and their interpretations*, DAIMI Report PB 78, Aarhus University, 1977

Retrieved from "https://en.wikipedia.org/w/index.php?title=Trace_monoid&oldid=929832823"

This page was last edited on 8 December 2019, at 14:49 (UTC).

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.