ACM PRESS

Proceedings of the
34th Annual ACM Symposium on Theory of Computing

Montreal, Quebec, Canada
May 19-21, 2002

Sponsored by

SIGACT

(The ACM Special Interest Group for Algorithms and Computation Theory)

ACM PRESS

Proceedings of the
34th Annual ACM Symposium on Theory of Computing

Montreal, Quebec, Canada
May 19-21, 2002

Sponsored by

SIGACT

(The ACM Special Interest Group for Algorithms and Computation Theory)

The Association for Computing Machinery
1515 Broadway
New York, New York 10036

Notice to Past Authors of ACM-Published Articles

ACM intends to create a complete electronic archive of all articles and/or other material previously published by ACM. If you have written a work that has been previously published by ACM in any journal or conference proceedings prior to 1978, or any SIG Newsletter at any time, and you do NOT want this work to appear in the ACM Digital Library, please inform permissions@acm.org, stating the title of the work, the author(s), and where and when published.

ACM ISBN: 1-58113-495-9

Additional copies may be ordered prepaid from:

ACM Order Department
PO Box 11405
New York, NY 10286-1405

Phone: 1-800-342-6626
(US and Canada)
+1-212-626-0500
(all other countries)
Fax: +1-212-944-1318
E-mail: acmhelp@acm.org

ACM Order Number 508020
Printed in the USA

Foreword

The papers in this volume were presented at the *Thirty-Fourth Annual ACM Symposium on Theory of Computing (STOC2002),* held in Montreal, Quebec, Canada, May 19-21, 2002. The Symposium was sponsored by the ACM Special Interest Group on Algorithms and Computation Theory (SIGACT).

In response to a call for papers, 287 paper submissions were received. All were submitted electronically. The program committee conducted its deliberations electronically, via an on-line meeting that ran from January 10 to January 19. The committee selected 91 papers from among the submissions. The submissions were not refereed, and many of these papers represented reports of continuing research. It is expected that most of them will appear in a more polished and complete form in scientific journals.

The papers encompassed in wide variety of areas of theoretical computer science. The topics included algorithms and computational complexity bounds for classical problems in algebra, geometry, topology, graph theory, game theory, logic and machine learning, as well as theoretical aspects of security, databases, information retrieval, and networks, the web, computational biology, and alternative models of computation including quantum computation and self-assembly.

The program committee would like to thank all authors who submitted papers for consideration. The committee is very grateful to the SIGACT Electronic Publishing Board for use of their software for the electronic meeting. The committee is especially grateful for the many colleagues listed below who helped us review the submissions.

We would also like to thank Lisa Tolles-Efinger of Sheridan Printing for her excellent aid in the proceedings production.

The Special Joint Session on Complexity Theory: This was held on May 21 jointly with the Seventeenth Annual IEEE Conference on Computational Complexity (COMPLEXITY2002).

The Knuth Prize: Christos Papadimitriou was awarded the Knuth Prize and gave a Plenary Talk on *"The Joy of Theory."*

The Machtey Award: TheSIGACT Best Student Paper was awarded to Tim Roughgarden for his submission *"The Price of Anarchy is Independent of the Network Topology."*

John Reif
Duke University
Program Chair, STOC2002

STOC'02 Conference Organization

SIGACT Chairman: Hal Gabow

Conference Program Chairman: John Reif, *Duke University*

Program Committee:

Pankaj Agarwal, *Duke University*

Alan Borodin, *University of Toronto*

Sam Buss, *University of California, San Diego*

Shafi Goldwasser, *Massachusetts Institute of Technology*

Rao Kosaraju, *John Hopkins University*

Richard Ladner, *University of Washington*

Gary Miller, *Carnegie Mellon University*

Michael Mitzenmacher, *Harvard University*

Satish Rao, *University of California, Berkeley*

John Reif, *Duke University*

Micha Sharir, *Tel-Aviv University*

Paul Spirakis, *University of Patris*

Mikkel Thorup, *ATT Research*

Leslie Valiant, *Harvard University*

Local Arrangement Co-Chairs:

Pierre McKenzie , *University of Montreal*

Denis Therien, *McGill University*

Publicity Chair: Ian Parberry

STOC'02 Reviewers

Dimitris Achlioptas, *Microsoft Research*
Micah Adler, *University of Massachusetts*
Susanne Albers, *Freiburg University*
Noga Alon, *Tel Aviv University*
Stephen Alstrup, *IT University of Copenhagen*
Andris Ambainis, *Institute for Advanced Studies, Princeton*
Amir Ben Amran, *Tel Aviv College*
David Applegate, *AT&T Labs-Research*
Aaron Archer, *Cornell University*
Lars Arge, *Duke University*
Albert Atserias, *Universitat Politecnica de Catalunya*
Hagit Attiya, *Technion*
Baruch Awerbuch, *Johns Hopkins University*
Yossi Azar, *Tel Aviv University*
Evripides Bampis, *University d'Evry*
Boaz Barak, *Weizmann Institute*
Amotz Bar-Noy, *Tel Aviv University*
Alexander Barvinok, *University of Michigan*
Paul Beame, *University of Washington*
Amos Beimel, *Ben-Gurion University*
Mihir Bellare, *University of California, San Diego*
Josh Benaloh, *Microsoft Research*
Michael Bender, *SUNY at Stony Brook*
Michael Benor, *Hebrew University*
Eli Ben-Sasson, *Harvard*
Guy Blellech, *Carnegie Mellon University*
Gerth S. Bordal, *BRICS*
Gilles Brassard, *University of Montreal*
Gerth Brodal, *BRICS*
Freddy Bruckstein, *Technion*
Hal Burch, *Carnegie Mellon University*
Ran Canetti, *Rutgers University/IBM*
Bernard Chazelle, *Princeton*
Alan Holun Cheng, *Duke University*
Benny Chor, *Tel Aviv University*
Richard Cleve, *University of Calgary*
Claude Crepeau, *University of McGill*
Artur Czumaj, *New Jersey Institute of Technology*
Tamal Dey, *Ohio State*
Joseph Diaz, *Universitat Politecnica de Catalunya*
Eleni Drinea, *Harvard*
Cynthia Dwork, *Microsoft, Inc.*
Martin Dyer, *University of Leeds*
Herbert Edelsbrunner, *Duke University*
Tuvi Etzion, *Technion*
Panagiota Fatourou, *University of Ionnina*
Uri Feige, *Weizmann Institute*
Amos Fiat, *Tel Aviv University*
Uriel Fiege, *Weizmann Institute*
Lance Fortnow, *NEC Research*
Dimitris Fotakis, *Max Planck Institute feur Informatik*
Nikolaos Foutoulakis, *Oxford University*
Pierre Fraignaud, *Universiti Paris-Sud*
Alan Frieze, *Carnegie Mellon University*
Michel Goemans, *Massachusetts Institute of Technology*
Andreas Goerdt, *Technische Universitaet Chemnitz*
Oded Goldreich, *Weizmann Institute*
Satish Govindarajan, *Duke University*
Love K. Grover, *Lucent Technologies*
Venkatesan Guruswami, *University of California at Berkeley*
Dan Gutfreund, *Hebrew University*
Dan Halperin, *Tel Aviv University*
John Harer, *Duke University*
Sariel Har-Peled, *University of Illinois at Urbana-Champaign*
Refael Hassin, *Tel Aviv University*
Maurice Herlihy, *Brown University*
Russell Impagliazzo, *University of California, San Diego*
Piotr Indyk, *Massachusetts Institute of Technology*
Kamal Jain, *Microsoft*
Klaus Jansen, *University of Kiel*
Mark Jerrum, *Edinburgh University*
David Johnson, *AT&T Labs-Research*
Valentine Kabenets, *University of California, San Diego*
Sampath Kannan, *University of Pennsylvania*
Bruce Kapron, *University of Victoria*
David Karger, *Massachusetts Institute of Technology*
Anna Karlin, *University of Washington*
Claire Kenyon, *University of Paris*
Roni Khardon, *Tufts University*
Joe Kilian, *Yianilos Labs*
Guy Kindler, *Tel Aviv University*
Jon Kleinberg, *Cornell University*
Adam Klivans, *Massachusetts Institute of Technology*
Spyros Kontogiannis, *Max Planck Institute feur Informatik*
Dimitris Koukopoulos, *Patras University*
Elias Koutsoupias, *University of Athens*
Jan Krajicek, *Charles University*
Hugo Krawczyk, *Technion*
Michael Krivelevich, *Tel Aviv University*
Eyal Kushilevitz, *Technion*

Jeff Lagarias, *AT&T*
Gadi Landau, *Haifa University*
Stefano Leonardi, *Universita' di Roma "La Sapienza"*
Leonid Levin, *Boston University*
Yehuda Lindell, *Weizmann Institute*
Phil Long, *Centrum voor Wiskunde en Informatica*
Tal Maklin, *ATT*
Christos Makris, *Patras University*
Yishay Mansour, *Tel Aviv University*
Yossi Matias, *Tel Aviv University*
Jiri Matousek, *Charles University*
Marios Mavronikolas, *University of Cyprus*
Colin McDiarmid, *Oxford University*
Frank McSherry, *University of Washington*
Kurt Mehlhorn, *Max Planck Institute feur Informatik*
Manor Mendel, *Hebrew University*
David Meyer, *University of California, San Diego*
Friedhelm Meyer auf der Heide, *University of Paderborn*
Silvio Micali, *Massachusetts Institute of Technology*
Daniele Micciancio, *University of California, San Diego*
Joseph Mitchell, *SUNY at Stony Brook*
Rajeev Motwani, *Stanford University*
S. Muthukrishnan, *AT&T*
Moni Naor, *Weizmann Institute*
Vijay Natarajan, *Duke University*
Minh Nguyen, *Harvard*
Sotiris Nikoletseas, *Patras University*
Kobbi Nisim, *DIMACS*
Noam Nissan, *Hebrew University*
Zeev Nutov, *Open University, Israel*
Joseph O'Rourke, *Smith College*
Rafail Ostrovosky, *Telcordia Technologies*
Janos Pach, *Renyi Institute*
Christos Papadimitriou, *University of California at Berkeley*
Vicky Papadopoulou, *Patras University*
David Parkes, *Harvard*
Mike Paterson, *University of Warwick*
David Peleg, *Weizmann Institute*
Pino Persiano, *University of Salerno*
Athanassios Poulakidas, *Patras University*
Cecilia Procopiuc, *AT&T*
Rajeev Raman, *University of Leicester*
Theis Rauhe, *IT University of Copenhagen*
Ran Raz, *Weizmann Institute*
Omer Reingold, *ATT*
Jennifer Rexford, *AT&T Labs-Research*
Justin Roberts, *University of California, San Diego*
Phili Rogaway, *UC Davis*
Dana Ron, *Tel Aviv University*
Amir Ronen, *Stanford University*
Alon Rosen, *Weizmann Institute*
Marie-Francoise Roy, *University of Rennes*
Mike Sacks, *Rutgers University*
Shmuel Safra, *Tel Aviv University*
Jared Saia, *University of Washington*
Alex Samorodnitsky, *Hebrew University*
Peter Sanders, *Max Planck Institute feur Informatik*
Christian Scheideler, *Johns Hopkins University*
Nate Segerlind, *University of California, San Diego*
Maria Serna, *Universitat Politecnica de Catalunya*
Ronen Shaltiel, *Weizmann Institute*
Nir Shavit, *Tel Aviv University*
Peter Shor, *AT&T Labs*
Amir Shplika, *Hebrew University*
Dan Simon, *Microsoft Research*
Alistair Sinclair, *University of California at Berkeley*
Mike Sipser, *Massachusetts Institute of Technology*
Steve Skiena, *SUNY at Stony Brook*
Danny Sleator, *Carnegie Mellon University*
Shakhar Smorodinsky, *Tel Aviv University*
Joel Spencer, *New York University*
Dan Spielman, *Massachusetts Institute of Technology*
Mike Steele, *University of Canterbury*
Ulrike Stege, *University of Victoria*
Madhu Sudan, *Massachusetts Institute of Technology*
Eva Tardos, *Cornell University*
Gabor Tardos, *Renyi Institute*
Michael Tarsi, *Tel Aviv University*
Amnon Ta-Shma, *Tel Aviv University*
Moshe Tennenholtz, *Technion*
Prasad Tetali, *Georgia Tech*
Luca Trevisan, *University of California at Berkeley*
Tom Trotter, *Arizona State University*
Salil Vadhan, *Harvard*
Kasturi Varadarajan, *University of Iowa*
Eric Vigoda, *Weizmann Institute*
Paul Vitanyi, *CWI, The Netherlands*
Paul Vitanyi, *University of Amsterdam*
Jeff Vitter, *Duke University*
Berthold Voecking, *Max Planck Institute feur Informatik*
Yusu Wang, *Duke University*
Tandy Warnow, *University of Texas at Austin*
Peter Winkler, *Bell Labs, Lucent*
Gerhard Woeginger, *University of Twente*
Neal Young, *Akamai Technologies*
Moti Yung, *CertCo*
Stathis Zachos, *National Technical University of Athens*
Christos Zaroliagis, *Computer Technology Institute*
Ofer Zeituni, *Technion*

Sponsors

The conference was sponsored by the ACM Special Interest Group on Algorithms and Computation Theory (SIGACT). The SIGACT Institutional Sponsors were:

Academic Press Inc.

Akamai Technologies

IBM Research

Microsoft Research

NEC Research Institute

STAR Lab, InterTrust Technologies Corp.

Table of Contents

Monday, May 20, 2002

Session 5A (Chair: Alan Borodin)

Session 5B (Chair: Michael Mitzenmacher)

Session 6A (Chair: Satish Rao)

Session 6B (Chair: Sam Buss)

Session 7A (Chair: Mikkel Thorup)

Session 7B (Chair: Gary Miller)

Session 8A (Chair: Sam Buss)

Session 8B (Chair: Pankaj Agarwal)

Tuesday, May 21, 2002

Session 9A (Joint Session with Complexity 2002) (Chair: Sam Buss)

Session 9B (Chair: Mikkel Thorup)

Session 10A (Joint Session with Complexity 2002) (Chair: Anne Condon)

Session 10B (Chair: John Reif)

Session 11A (Joint Session with Complexity 2002) (Chair: Michael Saks)

Session 11B (Chair: Pankaj Agarwal)

Session 12A (Joint Session with Complexity 2002) (Chair: Sam Buss)

Session 12B (Chair: Rao Kosaraju)

Recognizing String Graphs in NP

Marcus Schaefer* Eric Sedgwick† Daniel Štefankovič‡

ABSTRACT

A *string graph* is the intersection graph of a set of curves in the plane. Each curve is represented by a vertex, and an edge between two vertices means that the corresponding curves intersect. We show that string graphs can be recognized in **NP**. The recognition problem was not known to be decidable until very recently, when two independent papers established exponential upper bounds on the number of intersections needed to realize a string graph [18, 20]. These results implied that the recognition problem lies in **NEXP**. In the present paper we improve this by showing that the recognition problem for string graphs is in **NP**, and therefore **NP**-complete, since Kratochvíl [12] showed that the recognition problem is **NP**-hard. The result has consequences for the computational complexity of problems in graph drawing, and topological inference.

1. STRINGS, DRAWINGS, AND DIAGRAMS

A *string graph* is the intersection graph of a set of curves in the plane. A *(Jordan) curve*, or *string*, is a set homeomorphic to $[0, 1]$. Given a collection of curves $(C_i)_{i \in I}$ in the plane, the corresponding intersection graph is $(I, \{\{i, j\} : C_i \text{ and } C_j \text{ intersect}\})$. The *size* of a collection of curves is the number of intersection points (we assume that no three curves intersect in the same point). A graph isomorphic to the intersection graph of a collection of curves in the plane is called a *string graph.*

The string graph problem asks how string graphs can be recognized. The problem made its first explicit appearance in a 1966 paper by Sinden on circuit layout [21], although a similar question had been suggested earlier by Benzer on genetic structures [1]. The string graph problem was introduced to the combinatorial community by Ron Graham in 1976 [9].

From a combinatorial point of view we are interested in $c_s(G)$, the smallest number of intersections of a set of curves realizing a string graph. For graphs G that are not string graphs, we let $c_s(G)$ be infinity. With that we can define $c_s(n) = \max\{c_s(G) : G \text{ is a string graph on } n \text{ vertices}\}$. A computable upper bound on $c_s(n)$ implies decidability of the string graph problem. In 1991 Kratochvíl and Matoušek [14] showed rather surprisingly, that $c_s(n) \geq 2^{cn}$ for some constant c, and conjectured that $c_s(n) \leq 2^{cn^k}$ for some k. The papers by Pach and Tóth [18], and Schaefer and Štefankovič [20] established upper bounds of this form, implying decidability of the string graph problem in nondeterministic exponential time.

The string graph problem is closely related to a graph drawing problem, a connection we will make use of later. Given a graph $G = (V, E)$ and a set $R \subseteq \binom{E}{2} = \{\{e, f\} : e, f \in E\}$ on E, we call a drawing D of G in the plane a *weak realization* of (G, R) if only pairs of edges which are in R are allowed to intersect in D (they do not have to intersect, however). In this case we call (G, R) *weakly realizable.* We say that D is a *realization* of G if exactly the pairs of edges in R intersect in D.[1] Let us define $c_w(G, R)$ as the smallest number of intersections in a weak realization of (G, R), $c_w(G) = \max\{c_w(G, R) : (G, R) \text{ has a weak realization}\}$, and $c_w(m) = \max\{c_w(G) : G \text{ has } m \text{ edges}\}$.

The string graph problem can be reduced in polynomial time to the weak realizability problem [16, 12]. The reduction is as follows. Given a graph $G = (V, E)$, let $G' = (V \cup E, \{\{u, e\} : u \in e \in E\})$, and $R = \{\{\{u, e\}, \{v, f\}\} : \{u, v\} \in E\}$. Then G is a string graph if and only if (G', R) is weakly realizable.

In Theorem 4.4 we show that the weak realizability problem is in **NP**. Because of the reduction of the string graph problem to the weak realizability problem, and Kratochvíl's proof of **NP**-hardness of the string graph problem [12] this implies the following corollaries.

COROLLARY 1.1. *The string graph problem is complete for* **NP**.

COROLLARY 1.2. *The weak realizability problem is complete for* **NP**.

[1] Kratochvíl [13, 11, 12] calls (G, R) an *abstract topological graph*, and uses the word *feasible* for weakly realizable.

*mschaefer@cs.depaul.edu, Department of Computer Science, DePaul University, 243 South Wabash, Chicago, Illinois 60604, USA.

†esedgwick@cs.depaul.edu, Department of Computer Science, DePaul University, 243 South Wabash, Chicago, Illinois 60604, USA.

‡stefanko@cs.uchicago.edu, Department of Computer Science, University of Chicago, 1100 East 58th Street, Chicago, Illinois 60637, USA, and Department of Computer Science, Comenius University, Bratislava, Slovakia.

STOC'02, May 19-21, 2002, Montreal, Quebec, Canada.

The corollaries imply that the weak realizability problem can be reduced to the string graph problem in polynomial time. No natural polynomial time reduction witnessing this relationship is known (although there is an **NP**-reduction).

The weak realizability problem is a generalization of the concept of *crossing number* of a graph G, which is the smallest number of intersections necessary to draw G in the plane. Garey and Johnson showed that computing the crossing number is **NP**-complete [7]. Many variants of this problem have been considered in the literature, including the *pairwise crossing number* (or *crossing pairs number*), which is the smallest number of pairs of edges that need to intersect to draw G. Pach and Tóth recently showed that computing the pairwise crossing number is **NP**-hard [17]. Since there is an **NP**-reduction from this problem to the weak realizability problem, we have the following corollary.

COROLLARY 1.3. *The pairwise crossing number problem is complete for* **NP**.

The string graph problem is also related to Euler (or Venn) diagrams, and through these to topological inference. Given a specification of the relationships of concepts, such as "some A is B, some B is C, but no A is C", we can ask whether there is a diagram illustrating the relationship of the concepts (regions homeomorphic to the unit disk). In this particular case Figure 1 illustrates the given situation.

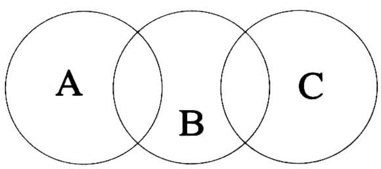

Figure 1: Some A is B, some B is C, but no A is C.

This problem is polynomial-time equivalent to the string graph problem. Topological inference allows a more refined set of predicates to describe relationship between regions, but even in this case a reduction to the string graph problem can be established, giving us the following result.

COROLLARY 1.4. *The existential theory of diagrams and the existential fragment of topological inference are complete for* **NP**.

Details of this reduction (which is an **NP**-reduction rather than a polynomial time one) and the definitions involved can be found in the journal version of [20]. Several restricted versions of this problem were shown to be solvable in **P** and **NP** earlier, but the general problem was not known to be decidable [10, 2, 22].

For the proof of our main theorem, Theorem 4.4, the same approach as in our earlier paper [20] proves successful: we reinterpret the problem as a problem over words. The necessary background material on words and word equations is covered in Section 2. The topological aspects of the proof are covered in Section 3.

2. WORD EQUATIONS

Let Σ be an alphabet of symbols and Θ be an alphabet of variables. The alphabets Σ and Θ are disjoint. A *word equation* $u = v$ is a pair of words $(u, v) \in (\Sigma \cup \Theta)^* \times (\Sigma \cup \Theta)^*$. The *size of the equation* $u = v$ is $|u| + |v|$. A *solution of the word equation* $u = v$ is a morphism $h : (\Sigma \cup \Theta)^* \to \Sigma^*$ such that $h(a) = a$ for all $a \in \Sigma$ and $h(u) = h(v)$ (h being a morphism means that $h(wz) = h(w)h(z)$ for any $w, z \in (\Sigma \cup \Theta)^*$). The *length of the solution* h is $\sum_{x \in \Theta} |h(x)|$.

A *word equation with specified lengths* is a word equation $u = v$ and a function $f : \Theta \to \mathbb{N}$. The solution h has to respect the lengths, i.e. we require $|h(x)| = f(x)$ for all $x \in \Theta$.

Let w be a word in Σ^*. We can write $w = c_1 f_1 c_2 \ldots c_k f_k$ where the c_i are characters in Σ, and the f_i are subwords of w. More precisely, $c_1 = w[1]$ and f_i is the longest prefix of $f_i c_{i+1} \ldots f_k$ which occurs in $c_1 f_1 \ldots f_{i-1} c_i$. The *Lempel-Ziv (LZ) encoding* of w is $LZ(w) = c_1[a_1, b_1]c_2 \ldots c_k[a_k, b_k]$ where $f_i = w[a_i \ldots b_i]$. The *size of the encoding* is $|LZ(w)| = k(\log|w| + \log|\Sigma| + 1)$. Note that some words can be compressed exponentially.

Let $h : (\Sigma \cup \Theta)^* \to \Sigma^*$ be a solution of an equation $u = v$. The LZ encoding of h is the sequence of LZ encodings of $h(x)$ for all $x \in \Theta$. The size of the encoding is $|LZ(h)| = \sum_{x \in \Theta} |LZ(h(x))|$. The usefulness of LZ encoding for word equations is demonstrated by following two results.

THEOREM 2.1 ([8]). *Let $u = v$ be a word equation. For an LZ encoding of a morphism h we can check whether h is a solution of the equation in time polynomial in $|LZ(h)|$.*

THEOREM 2.2 ([19]). *Let $u = v$ be a word equation with lengths specified by a function f. Assume that $u = v$ has a solution respecting the lengths given by f. Then there is a solution h respecting the lengths such that $|LZ(h)|$ is polynomial in the size of the binary encoding of f and the size of the equation. Moreover, the lexicographically least such solution can be found in polynomial time.*

Given an equation with specified lengths there might be solutions which can not be LZ compressed. However Theorem 2.2 says that there is a solution which can be LZ compressed. In particular if the equation has a unique solution then that solution can be LZ compressed. Note that it is easy to encode several equations into one equation [15, Proposition 12.1.8], hence Theorems 2.1 and 2.2 hold for systems of equations as well.

We will need the following two results which easily follow from [8].

LEMMA 2.3. *For an LZ-encoding $LZ(w)$ of w we can test whether w is a palindrome in time polynomial in $|LZ(w)|$.*

LEMMA 2.4. *Given an LZ encoding $LZ(w)$ of w and $a \in \Sigma$, we can compute the number of occurrences of a in w in time polynomial in $|LZ(w)|$.*

3. COMPUTATIONAL TOPOLOGY

In the following let M be a compact orientable surface with boundary. A simple arc γ such that both its endpoints $\gamma(0), \gamma(1)$ are on the boundary ∂M and the internal points $\gamma(x), 0 < x < 1$ are in the interior $\dot{M}$ is called a *properly embedded arc.* Two properly embedded arcs γ_1, γ_2 are *isotopic rel. boundary* ($\gamma_1 \sim \gamma_2$) if there is a continuous deformation of γ_1 to γ_2 which does not move the endpoints. The *isotopy class* of γ is the set of properly embedded arcs isotopic to