Chapter 12

Lecture 12

Advanced Topics in Graph Algorithms		
©Tel-Aviv University		
Spring Semester, 1994		
Lecturer	:	`Ron Shamir`
Scribe	:	`Michael Seltser`
Lecture 12	:	`June 30`

12.1 Temporal Reasoning and Interval algebras

In this section we deal with problems in temporal reasoning. Let us start with an example.

Example 12.1 *A paleonthologist wants to determine the relative span of existence of the species Brontosaurus, Parasaurulophus, Stegosaurus, and Tyrannosaurus. His assumptions:*

- *B and P did not exist simultaneously*
- *P perished before S emerged*
- *S emerged before T and perished not after it*
- *Either P perished before T emerged or P emerged when T perished*
- *B emerged and perished during T's existence*

Is there a scenario satisfying all his assumptions ?

See figure 12.1 for a possible scenario consistent with the assumptions.

Allen in 1983 has defined a model for temporal logic whose elements are time intervals (corresponding to events that happen during a certain period of time) and used the following 13 primitive relations:

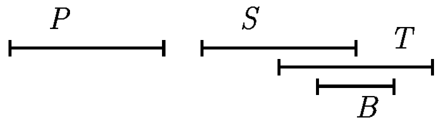

Figure 12.1: A consistent scenario

$$\{\prec, \succ, m, m^{-1}, o, o^{-1}, s, s^{-1}, f, f^{-1}, d, d^{-1}, \equiv\}$$

to express the relative position of two intervals (See Figure 12.2). We will call this *the 13-valued interval algebra* $\mathcal{A}_{13}$.

RELATION	NOTATION	INTERPRETATION
x before y	$\prec$	
y after x	$\succ$	
x meets y	m	
y met-by x	m^{-1}	
x overlaps y	o	
y overlapped-by x	o^{-1}	
x starts y	s	
y started-by x	s^{-1}	
x during y	d	
y includes x	d^{-1}	
x finishes y	f	
y finished-by x	f^{-1}	
x equals y	$\equiv$	

Figure 12.2: The 13-valued interval algebra $\mathcal{A}_{13}$. Single line: x interval. Double line: y interval.

From these we define the 3-valued interval algebra $\mathcal{A}_3$ whose elements are called its *atomic relations*

$$\mathcal{A}_3 \; : \; \{\prec, \succ, \cap\}$$

Here $X \cap Y$ denotes that events (intervals) x and y have a non-empty intersection, i.e., any one of the eleven relations excluding $\prec$ and $\succ$ in Allen's algebra may hold between x and y. The choice of which algebra to use depends on the nature of the application.

12.2 Interval satisfiability problems

We now define our temporal reasoning problems in the context of constraint satisfiability. For each pair of events x and y, let $D(x,y)$ be a set of atomic relations in the algebra $\mathcal{A}_i$. The semantics here is that we do not know precisely the relationship between x and y, but it must be one of those in the set $D(x,y)$. (In the language of constraint satisfiability, there is a variable $v(x,y)$ representing the relation between x and y, and its value must be taken from those in the set $D(x,y)$ corresponding to its domain.)

For example, we read $D(x,y) = \{\prec, o\}$ as x is either before or overlaps y. We also call a set of atomic relations a *relation set.* Occasionally we shall use the notation $xD(x,y)y$ for short. For example, $D(x,y) = \{\prec, d^{-1}, \equiv\}$ and $x\{\prec, d^{-1}, \equiv\}y$ both mean "either x ends before y starts, or y is during x, or the two are equal". We omit braces and commas when there is no ambiguity, e.g., $x{\prec}y$ or $x{\cap}y$.

In particular, relation sets in $\mathcal{A}_3$ will be presented by concatenation of the atomic relations. Hence, $x{\prec}{\succ}y$ denotes x and y are disjoint, $x{\prec}{\cap}y$ denotes $x\{\prec, \cap\}y$, etc.

An *instance* (or *input*) for all the problems studied in this paper consists of n events, and a relation set $D(x,y)$ for every pair of distinct events x, y. Without loss of generality, we assume that for each pair of events x and y, the relation sets $D(x,y)$ and $D(y,x)$ given as input are consistent, i.e., for each atomic relation R, $R \in D(x,y) \Leftrightarrow R^{-1} \in D(y,x)$. Otherwise, it is a simple matter to restrict the relation sets further so that they satisfy these properties or are shown to be unsatisfiable. Hence, we can assume that the input contains for each pair of events x, y only one relation set, say $D(x,y)$, and that the other relation set $D(y,x)$ is given implicitly by the above rule.

Remark 12.2 *The* size *of an instance is the number s of non-trivial relation sets in it, and is linearly equivalent to the binary input length for each $\mathcal{A}_i$. We will assume that relation sets which are trivial (i.e., contain all the atomic relations) are not listed explicitly as part of a problem input, and that each event is involved in at least one non-trivial relation set.*

Hence, the input size s of a problem with n events satisfies $s = O(n^2)$ and $s = \Omega(n)$.

An *interval realization* for the instance $\{D(x,y)\}$ is an assignment of intervals on the real line to events, so that for each pair of events, one of the atomic relations in their relation set holds. Since the algebras discussed here are concerned with topological properties only, a realization can be viewed also as a complete weak order of the interval endpoints, thereby identifying all realizations which differ in metric only. Two realizations are *distinct* if the orders of the endpoints in them differ. The input data $\{D(x,y)\}$ is *consistent* (or *satisfiable*) if it admits at least one realization.

The *interval satisfiability problem* (ISAT) is determining the existence of (and finding) one instantiation that is consistent with the input data $\{D(x,y)\}$. Occasionally, we shall make an additional distinction between the *decision version* of ISAT, whose output is only a yes/no answer, and the *constructive version*, whose output is the answer 'no' if the input is inconsistent and a solution if it is consistent.

The *minimal labeling problem* (MLP) is to determine the minimal sets $D'(x,y) \subseteq D(x,y)$ such that the set of realizations is unchanged, and every remaining atomic relation participates in some solution.

Example 12.3 $x\{\prec, m, o\}y$, $y\{\prec, \equiv, \succ\}z$, $z\{f, s\}x$.
Here xoy, $y{\succ}z$, zsx, *and* $x{\prec}y$, $y{\succ}z$, zfx *are both consistent with the input, as shown in Figure 12.3. On the other hand,* $y{\prec}z$ *and* $z \equiv y$ *are impossible. The minimal labeling for this problem is* $x\{\prec, m, o\}y$, $y{\succ}z$, $z\{f, s\}x$.

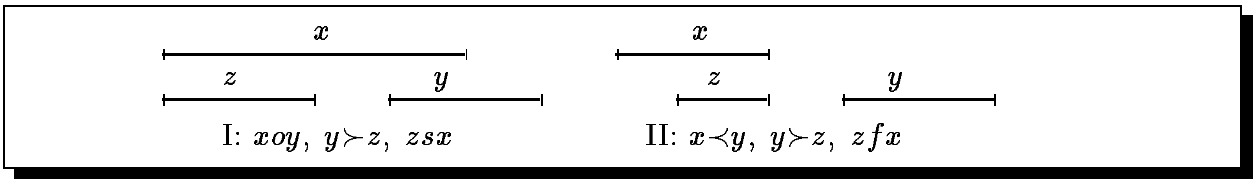

Figure 12.3: Two interval realizations for Example 12.2

The *all-consistent-realizations problem* (ACRP) is finding of all the consistent realizations. Here there may be exponentially many solutions, but we would like to generate them in time polynomial per each new realization.

A *fragment* of an algebra ($\mathcal{A}_3$ or $\mathcal{A}_{13}$) is a collection of relation sets in that algebra. A problem is restricted to a fragment F if all relation sets participating (implicitly or explicitly) in the instance belong to F. We will denote the problem A restricted to fragment F by $A(F)$.

Example 12.4 *In the problem ISAT* $(\prec, \succ)$, *for every pair of events* x, y *either* $x{\prec}y$ *or* $x{\succ}y$. *Clearly the input is consistent iff the relation* $\prec$ *is a linear order, as no intersections are allowed. This can be checked in linear time.*

Example 12.5 *ISAT* $(\prec, \succ, \prec\succ)$ *is polynomial by a similar argument. Here the instance is consistent iff* $\prec$ *is a partial order.*

Example 12.6 *ISAT* $(\prec\succ, \cap)$ *is equivalent to interval graph recognition: Form a graph* G *with a vertex for each event with an edge between* x *and* y *iff* $x{\cap}y$. *The instance has a realization iff* G *is an interval graph. Hence the problem is solvable in linear time by the algorithm of Booth and Leuker (1976).*

Theorem 12.7 (Golumbic - Shamir, 93) *For each of* $\mathcal{A}_{13}, \mathcal{A}_3$, *MLP and ACRP are polynomially equivalent to ISAT.*

J.Allen defined a constraint propagation polynomial heuristic for solving ISAT on $\mathcal{A}_{13}$. If the algorithm returns a negative answer then it is correct, but a positive answer may be incorrect. He conjectured that ISAT is NP-complete on $\mathcal{A}_{13}$. This was proved by Vilain and Kautz (1986). Vilain, Kautz and van Beek (1989) have shown that Allen's algorithm always gives the correct answer if the input data is restricted to a fragment of $\mathcal{A}_{13}$. Nökel (1988) has given another polynomial algorithm on a different fragment of $\mathcal{A}_{13}$.

Applications of interval satisfiability problems are numerous. They include planning, molecular biology (DNA), knowledge representation, natural language processing, information systems (databases), artificial intelligence, behavioral psychology, archaeology (seriation), graph theory and others.

12.3 Interval sandwich problems and ISAT

Let E^1 and E^2 be two disjoint sets of edges defined on the same vertex-set V. A graph $G(V, E)$ with $E^1 \subseteq E \subseteq E^1 \cup E^2$ is called a *sandwich graph* for (E^1, E^2). The interval graph sandwich (IGS) problem is the following:

Interval Graph Sandwich problem:

INPUT: Two graphs $G^1 = (V, E^1)$ and $G^2 = (V, E^2)$ with the same set of vertices, and disjoint edge-sets E^1 and E^2.

QUESTION: Is there a sandwich graph for (E^1, E^2) which is an interval graph?

Define $F = \overline{\{E^1 \cup E^2\}}$ (i. e., F is the set of non-edges in the graph $(V, E^1 \cup E^2)$.) When $E^1 = \emptyset$ or $F = \emptyset$, the answer is trivially yes. When $E^2 = \emptyset$, the problem is equivalent to recognizing if $G(V, E^1)$ is interval and thus is polynomial by the algorithm of Booth and Lueker. We shall show that in the general case, the problem is NP-complete.

Given a set of intervals on the real line, one can define a partial order on the intervals by $a \prec b$ if and only if the interval a is completely to the left of interval b. We use this *interval order* to give a reduction from the following problem:

BETWEENNESS:

INPUT: A set of elements $S = \{a_1, \ldots, a_n\}$, and a set $T = \{T_1, \ldots, T_m\}$ of ordered triplets of elements from S, where $T_i = (a_{i_1}, a_{i_2}, a_{i_3})$ $i = 1, \ldots, m$.

QUESTION: Does there exist a one-to-one function $f : S \to \{1, 2, \ldots, n\}$ such that either $f(a_{i_1}) < f(a_{i_2}) < f(a_{i_3})$ or $f(a_{i_1}) > f(a_{i_2}) > f(a_{i_3})$ for $i = 1, \ldots, m$?

The problem was shown to be NP-complete by Opatrny (1980).

Theorem 12.8 *The interval graph sandwich problem is NP-complete.*

Proof: Membership in NP follows since a given interval realization can be checked to fit the data in polynomial time. Given an instance of BETWEENNESS, we transform it to an interval sandwich instance as follows: Define a vertex v_i for each element a_i, and two vertices x_j^1 and x_j^2 for triplet T_j. The vertex set is $V = \{v_1, \ldots, v_n\} \cup \{x_j^1, x_j^2 \mid j = 1, \ldots, m\}$. The edges sets are

$$E^1 = \{(v_{i_1}, x_i^1), (x_i^1, v_{i_2}),$$
$$(v_{i_2}, x_i^2), (x_i^2, v_{i_3}) \mid i = 1, \ldots, m\}$$
$$F = \{(v_i v_j) \mid i \neq j\} \cup$$
$$\{(v_{i_1}, x_i^2), (v_{i_3}, x_i^1), (x_i^1, x_i^2) \mid i = 1, \ldots, m\}$$

In other words, to each triplet corresponds a 5-chain (see figure 12.4(a)) which should also appear as a 5-chain in the sandwich graph. In addition, all the v_i-s should form an independent set in that graph. The reduction is clearly polynomial.

The key to the reduction is the following simple observation: in any interval realization of a 5-chain (a, x, b, y, c), either $a \prec b \prec c$ or $c \prec b \prec a$ (see figure 12.4(b)).

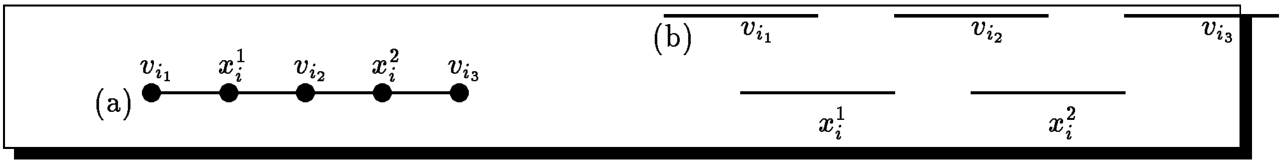

Figure 12.4: (a) The 5-chain $(v_{i_1}, x_i^1, v_{i_2}, x_i^2, v_{i_3})$. (b) An interval realization of the 5-chain.

Suppose there is an ordering f of the set S which satisfies the betweenness conditions. Create an interval realization for an interval sandwich as follows: Draw n disjoint intervals corresponding to the v_i-s in the same order as the a_i-s appear in f. That is, $I(v_i) \prec I(v_j)$ if and only if $f(a_i) < f(a_j)$. Since the order conditions on triplets are satisfied, additional intervals corresponding to x_i^1 and x_i^2 can be added for each triplet T_i such that the subgraph induced by the five vertices $\{v_{i_1}, v_{i_2}, v_{i_3}, x_i^1, x_i^2\}$ is a 5-chain. The resulting interval graph on V is a sandwich graph.

For the converse, suppose there exists an interval sandwich solution with realization I. Since in that graph the set $\{v_1, \ldots, v_n\}$ induces an independent set, a complete ordering on S is generated: Number the elements in the order of their intervals from left to right, namely, $f(a_i) < f(a_j)$ if and only if $I(v_i) \prec I(v_j)$. Since for each triplet T_i the five vertices $\{v_{i_1}, v_{i_2}, v_{i_3}, x_i^1, x_i^2\}$ must induce a 5-chain in every sandwich graph, by the above observation, every triplet satisfies the betweenness condition. ∎

Interestingly, we can also show that a restricted version of the above problem is NP-complete: A function $c : V \to Z$ is called a *coloring* of the graph $G = (V, E)$ if for every $(u, v) \in E$, $c(u) \neq c(v)$. The problem is defined as follows:

COLORED INTERVAL SANDWICH:
INPUT: A graph $G = (V, E)$ and a coloring c of G.
QUESTION: Does there exist a set E' so that $E \subseteq E'$, $G' = (V, E')$ is an interval graph and c is also a coloring for G'?

Notice that this is a special case of the sandwich problem where $E^1 = E$ and $F = \{(u, v) \mid c(u) = c(v)\}$ is given implicitly by the coloring.

Theorem 12.9 *The colored interval sandwich problem is NP-complete.*

Proof: Reduction from BETWEENNESS, where the vertex set and $E = E^1$ is defined as in Theorem 12.8 and the coloring is $c(x_i^1) = c(x_i^2) = i$ $(i = 1, \ldots, m)$ and $c(v_1) = c(v_2) = \cdots = c(v_n) = m + 1$. A slight modification of the arguments in Theorem 12.8 proves the result. ∎

Remark 12.10 *Golumbic, Kaplan, and Shamir (1993) have shown that the analog problems to Theorem 12.8 and 12.9, with the additional condition that all intervals have the same length, are still NP-complete.*

Corollary 12.11 *ISAT($\prec\succ, \cap, \prec\cap\succ$) is NP-complete.*
Proof: *Just consider $V = \{$ events $\}$, $(i, j) \in E^1 \Leftrightarrow i \cap j$, $(i, j) \in F \Leftrightarrow i \prec\succ j$, $(i, j) \in E^2 \Leftrightarrow i \prec\cap\succ j$.* ∎

Corollary 12.12 *ISAT($\mathcal{A}_3$) is NP-complete.*
Proof: *Follows from Corollary 12.11 since ISAT($\prec\succ, \cap, \prec\cap\succ$) is a restriction of ISAT($\mathcal{A}_3$).* ∎

Corollary 12.13 *ISAT, MLP, ACRP are NP-hard for $\mathcal{A}_3$ and $\mathcal{A}_{13}$.*

Theorem 12.14 *ISAT($\prec$, $\succ$, $\prec\succ$, $\cap$) is polynomial.*

Proof: exercise. ∎

12.4 A linear time algorithm for ISAT(Δ_1)

In this section we deal with ISAT problems on the fragment:

$$\Delta_1 = \{\prec, \succ, \cap, \prec\cap, \cap\succ, \prec\cap\succ\}.$$

That is, $\mathcal{A}_3 - \prec\succ$. We shall give efficient algorithms for ISAT, MLP and ARP on this fragment, and they will apply immediately to any subfragment of Δ_1. Hence, by excluding just a *single* relation set from $\mathcal{A}_3$ the problems become tractable.

The polynomial algorithm for ISAT(Δ_1) utilizes a directed graph, which will be described later, with vertices corresponding to the endpoints of event intervals and arcs representing the relative order of endpoints. The key observation here is that every relation in Δ_1 is equivalent to a certain *order requirement* between a pair (or two pairs) of such endpoints. This observation is proved below:

Lemma 12.15 *Let i and j be the event intervals $[l_i, r_i]$ and $[l_j, r_j]$, respectively. In each of the following cases, the intervals satisfy the set of relations if and only if their endpoints satisfy the corresponding inequalities:*

$$i{\prec}j \;\Leftrightarrow\; r_i < l_j \qquad\qquad i{\succ}j \;\Leftrightarrow\; r_j < l_i \tag{12.1}$$

$$i{\prec}{\cap}j \;\Leftrightarrow\; l_i \leq r_j \qquad\qquad i{\cap}{\succ}j \;\Leftrightarrow\; l_j \leq r_i \tag{12.2}$$

$$i{\cap}j \;\Leftrightarrow\; l_i \leq r_j \text{ and } l_j \leq r_i \tag{12.3}$$

If $i{\prec}{\cap}{\succ}j$, no constraint is imposed.

Proof: (1) is immediate. The conditions in (2) are the negation of those in (1). (3) is equivalent to the intersection of the two conditions in (2). ∎

The graph is now constructed as follows: For an instance J of ISAT(Δ_1) with n events, form a directed graph $G(J) = G(V, E)$ with vertex set $V = \{r_1, \ldots r_n, l_1, \ldots l_n\}$. The arc set E consists of two disjoint subsets, E_0 and E_1.

The former will represent weak inequality and the latter will represent strict inequality between pairs of endpoints.

The arcs are defined as follows:

$$(l_i, r_i) \in E_0 \qquad\qquad i = 1, \ldots, n \tag{12.4}$$

$$(r_i, l_j) \in E_1 \qquad\qquad \forall i, j \text{ s.t. } i{\prec}j \tag{12.5}$$

$$(l_i, r_j) \in E_0 \qquad\qquad \forall i, j \text{ s.t. } i{\prec}{\cap}j \tag{12.6}$$

$$(l_i, r_j) \in E_0 \text{ and } (l_j, r_i) \in E_0 \qquad\qquad \forall i, j \text{ s.t. } i{\cap}j \tag{12.7}$$

For pairs i, j with the relation $i{\prec}{\cap}{\succ}j$, no arc is introduced. Define now $E = E_0 \cup E_1$. We call the arcs in E_0 (resp., E_1) the *weak arcs* (resp., *strict arcs*). Note that the graph G is bipartite. Denote the two parts of the vertex set by $R = \{r_1, \ldots, r_n\}$ and $L = \{l_1, \ldots, l_n\}$, and call an arc an *RL-arcs* (resp., *LR-arc*) if it is directed from R to L (resp., from L to R).

Remark 12.16 *In the graph G all the RL-arcs are strict and all the LR-arcs are weak. Hence, we need not record explicitly the type of each arc since it is implied by its direction.*

Lemma 12.17 *Every cycle in G contains a strict arc.*

Algorithm: Δ_1-*CONSISTENCY*
begin
1. construct $G(J)$ according to rules (12.4) - (12.7).
2. if $G(J)$ contains a cycle - J is not satisfiable. Otherwise, it is.

end *construct G-decomposition*

Figure 12.5: Algorithm to construct $Delta_1$-CONSISTENCY

Proof: Since G is bipartite, a cycle must contain vertices both from R and from L. In particular, it must contain an RL-arc, so the claim follows by Remark 12.16. ∎

An algorithm for solving ISAT(Δ_1) is described in figure 12.5

Lemma 12.18 *Suppose $G(J) = (V, E)$ is acyclic. Then a linear order on V is consistent with the partial order $G(J)$ if and only if it is a realization of J.*

Proof: Take any linear order P which extends the partial order G. P is an ordering of all the endpoints, which by Lemma 12.15 satisfies all the relations in the input, so P gives a realization for J. On the other hand, every realization of J gives a linear order of the endpoints, in which each of the input relations must be satisfied, by Lemma 12.15. Hence, the linear order must be consistent with the partial order $G(J)$. ∎

Theorem 12.19 *Algorithm Δ_1-$CONSISTENCY$ correctly recognizes if an instance of ISAT(Δ_1) is satisfiable in linear time.*

Proof: Validity: By Lemma 12.15, each arc reflects the order relation of a pair of interval endpoints as prescribed by the input relations.

If G contains a cycle, then by Lemma 12.17 that cycle contains a strict arc. Hence, that cycle must satisfy $r_i < l_j \leq \ldots \leq r_i$, which implies that the input relations cannot be satisfied. In case G is acyclic,

Lemma 12.18 implies that J is satisfiable.

Complexity: Constructing $G(J)$ requires $O(s)$ steps, where s is the number of input relation sets, since the effort is constant per relation. Checking if G is acyclic can be done, for example, by depth first search, in time linear in $|E|$, the number of arcs.

Since $|E| \leq 2s + n$, using remark 12.2 we conclude that the algorithm is linear. ∎

In a similar way one can construct algorithms for MLP(Δ_1) and ACRP(Δ_1) in the following manner: An algorithm for MLP is obtained by calculation of transitive closure of G. The algorithm for ACRP(Δ_1) is based on finding of all linear extensions of G.

Remark 12.20 *ISAT is polynomial if all events are points ($\mathcal{A}_3 = \mathcal{A}_{13}$ in that case).*

Proof: Exercise. ∎

Remark 12.21 *One can handle additional metric constraints in $\mathcal{A}_3 - \prec\succ$, by solving the linear programming problem defined by (12.1) - (12.3) with additional metric constraints on the endpoints.*

		$\cap$	$\prec\succ$	$\prec\succ, \cap$
	–	T	T	Poly interval graph
$\prec$	Poly linear order	Poly interval order	Poly acyclic graph	Poly
$\prec\cap$	T	T	NPC*	NPC*
$\prec\cap\succ$	T	T	T	NPC interval sandwich
$\prec, \prec\cap$	Poly	Poly	NPC*	NPC*
$\prec, \prec\cap\succ$	Poly	Poly	Poly	NPC
$\prec\cap, \prec\cap\succ$	T	T	NPC	NPC
$\prec, \prec\cap, \prec\cap\succ$	Poly	Poly	NPC	NPC

Figure 12.6: Complexity of ISAT on all fragments of $\mathcal{A}_3$. T: satisfied by setting all intervals disjoint or identical. The four asterisk cases were shown to be NPC by Webber (1994).

Figure 12.6 summarizes results from Golumbic - Shamir (1993) and Webber (1994) on the complexity of ISAT on all fragments of $\mathcal{A}_3$.

12.5 Bandwidth, Pathwidth and Proper Pathwidth

We now switch to different problems related to interval and proper interval graphs. The motivation studying to these problems arises in physical mapping of DNA, a major problem in the Human Genome Project.

Lemma 12.22 *Every interval graph (and every proper interval graph) has an interval (resp., proper interval) representation on the real line in which all endpoints are distinct, and the left endpoints are assigned to the integers* $1, 2, \ldots, |V|$.

We shall call such a representation *canonical.*

Definition Pathwidth and Proper Pathwidth A *path decomposition* of a given graph $G = (V, E)$, is a sequence of subsets of V, $X = (X_1, \ldots, X_l)$ such that

(1) $V = \cup_i X_i$

(2) For each edge $(u, v) \in E$, there exists some $i \in \{1, \ldots, l\}$ so that both u and v belong to X_i.

(3) For each $v \in V$ there exist some $s(v), e(v) \in \{1, \ldots, l\}$ so that $s(v) \leq e(v)$, and $v \in X_j$ if and only if $j \in \{s(v), s(v) + 1, \ldots, e(v)\}$.

The *width* of X is defined by $pw_X(G) = \max\{|X_i| \mid i = 1, \ldots, l\} - 1$. The *pathwidth* of G, denoted $pw(G)$, is the minimum value of $pw_X(G)$ over all path decompositions, i.e., $pw(G) = \min\{pw_X(G) \mid X \text{ is a path decomposition of } G\}$. The notion of pathwidth was originally introduced by Robertson and Seymour (1983).

Lemma 12.23 (e.g., Mohring, 91) *For any path decomposition $X = (X_1, \ldots, X_l)$ of a graph G, every clique in G must be contained in some X_i.*

The *PATHWIDTH* problem is to decide for a given graph G and a given integer k if $pw(G) \leq k$. Equivalent problems arise in various areas, including VLSI layout, processor management, node searching and vertex separation.

Lemma 12.24 *For every graph G, the pathwidth of G is one less than the least clique size of any interval supergraph of G.*

We introduce here the notion of a *proper path decomposition* of G; It is as a path decomposition X which satisfies (1)-(3) and also:

(4) For every $u, v \in V$, $\{s(u), s(u) + 1, \ldots, e(u)\} \not\subset \{s(v), s(v) + 1, \ldots, e(v)\}$.

(As usual, $\subset$ denotes strict containment.) The *proper pathwidth* of G, denoted $ppw(G)$, is the minimum value of $pw_X(G)$ over all proper path decompositions of G, namely $ppw(G) = \min\{pw_X(G) \mid X \text{ is a proper path decomposition of } G\}$.

Clearly, $pw(G) \leq ppw(G)$, but note that $pw(G)$ and $ppw(G)$ can differ dramatically: For the star graph with $2n + 1$ vertices, $G = K_{1,2n}$, $pw(G) = 1$ but $ppw(G) = n$. An easy way to see the last equality is using the characterizations we shall provide in the next section.

Bandwidth: Another measure of graph width which we shall use is bandwidth: For $G = (V, E)$ with $|V| = n$, a linear *layout* of the graph is a 1-1 function $L : V \to \{1, \ldots, n\}$. The *bandwidth of L* is $bw_L(G) = \max_{(u,v) \in E}\{|L(u) - L(v)|\}$. The *bandwidth* of G is the minimum bandwidth over all layouts of G, namely, $bw(G) = \min\{bw_L(G) \mid L \text{ is a layout of } G\}$. The *BANDWIDTH* problem is to decide for a given graph G and integer k, if $bw(G) \leq k$. This problem has been studied intensely because of its application to sparse matrix algebra. It is known to be NP-complete even for binary trees (Garey, Graham, Johnson, and Knuth, 1978) and for caterpillars with hair length at most three (Monien, 1986). On the other hand, it is solvable in $O(n^k)$ for arbitrary k and in linear time for $k = 2$ (Garey, Graham, Johnson, and Knuth, 1978).

Lemma 12.25 (Kaplan - Shamir, 94) *For every graph G, the proper pathwidth of G is one less than the least clique size of any proper interval supergraph of G.*

Proof: Suppose $X = (X_1, \ldots, X_l)$ is a proper path decomposition of G and $pw_X(G) = ppw(G) = k$. On the linear order $1 < 2 < \cdots < l$, let $I(v)$ be the interval $[s(v), e(v)]$. Then $\{I(v)\}_{v \in V}$ is a representation of an interval graph G'. G' is proper since no interval is strictly contained in the other, by property (4) in the definition of a proper path decomposition. G' is a supergraph of G, by property (2). Since X is also a path decomposition of G', Lemma 12.23 implies that every clique in G' must be contained in some X_i. Hence, $\omega(G') \leq k + 1$.

Conversely, suppose $k + 1$ is the least clique size of any proper interval supergraph of G. Let $\{I(v)\}_{v \in V}$ be a proper interval representation of such supergraph $G' = (V, E')$ with $\omega(G') = k + 1$, in which all endpoints are distinct, and $I(v) = [l(v), r(v)]$. Order the 2n endpoints from left to right $p_1, \ldots p_{2n}$. Define $X_i = \{v \in V \mid p_i \in I(v)\}$ for $i = 1, \ldots, 2n$. (Here and throughout $n = |V|$). We claim that $(X_1, \ldots, X_{2n})$ is a proper path decomposition of G. Checking the four requirements from a proper path decomposition one can observe that (1) is immediate. (2) follows since $E \subseteq E'$, and if $(u, v) \in E'$ and $l(u) < l(v)$ then $l(v) \in I(v) \cap I(u)$. In other words, every edge in E' is represented in some X_i. Since G' is a proper interval graph, it follows that $s(v) = l(v)$ and $e(v) = r(v)$ satisfy (3) and (4). Since the clique size of G' is $k + 1$, each point p_i can meet at most $k + 1$ intervals in the representation. Hence, $\max_i |X_i| \leq k + 1$ so $ppw(G) \leq k$. ■

Theorem 12.26 (Kaplan - Shamir, 94) *The bandwidth of a graph equals its proper pathwidth.*

Proof: For the graph $G = (V, E)$ with $ppw(G) = k$, by Lemma 12.25 there exists a supergraph $G' = (V, E')$ of G which is proper interval with clique size $k + 1$. Let $\{I(v)\}_{v \in V}$ be a canonical representation for G', where $I(v) = [l(v), r(v)]$. Define a layout L on G by $L(u) = l(u)$ for all $u \in V$. Suppose $L(u) - L(v) > k$ for some $(u, v) \in E$, where $L(v) = i$. Then $\{L^{-1}(i), \ldots, L^{-1}(i + k + 1)\}$ form a clique of size $k + 2$ in G', a contradiction. Hence, $bw(G) \leq bw_L(G) \leq ppw(G)$.

Conversely, let L be a layout of G so that $bw_L(G) = bw(G) = k$. Form a set of equal-length intervals on the real line $\{I(v)\}_{v \in V}$ where $I(v) = [L(v), L(v) + k]$. Let $G' = (V, E')$ be the intersection graph of that set of intervals. Clearly $E' \supseteq E$ since if $(u, v) \in E$ then $|L(u) - L(v)| \leq k$, so $I(u) \cap I(v) \neq \emptyset$. Since the maximum clique size for G' is at most $k + 1$, using Lemma 12.25, $ppw(G) \leq k$. Hence, $bw(G) \geq ppw(G)$. ■

This interesting - and somewhat surprising - equivalence between bandwidth and proper pathwidth allows us to draw several immediate conclusions from the results on bandwidth:

Corollary 12.27

A. *Finding the proper pathwidth is NP-hard, even for binary trees and for caterpillars with*

hair length at most three.
B. *Problem A can be solved in* $O(f(k)n^{k-1})$ *time, and in particular is polynomial when* k *is fixed.*
C. *Deciding whether the proper pathwidth of a graph is not greater than two can be done in linear time.*

These results follow immediately from known results on bandwidth, using Theorem 12.26.

We finish by mentioning two parametric results. The first deals with the unit interval sandwich problem mentioned above:

Theorem 12.28 (Kaplan-Shamir 94) *Deciding if a sandwich instance admits a proper interval sandwich graph with clique size at most* k *can be done in* $O(C_k n^k)$ *time, and in particular in polynomial time when* k *is fixed. Furthermore, it cannot be solved in* $O(C_k n^\alpha)$ *where* α *is independent of* k*, unless deciding if a given graph has an independent set of size* k *can be solved in* $O(C_k' n^{\alpha'})$.

In contrast to that negative result, the completion problem for proper interval graphs is tractable, in the parametric sense:

Theorem 12.29 (Kaplan, Shamir, Tarjan 94) *Given a graph* $G = (V, E)$*, one can decide if there exists a proper interval graph* $G' = (V, E')$ *with* $E \subseteq E'$ *and* $|E' \setminus E| \leq k$ *in time* $O(C_k(|V| + |E|))$.