Document classes with decidable XPath satisfiability

Paweł Parys
Warsaw University

Mikołaj Bojańczyk, Diego Figueira, Vince Barany

Satisfiability problem:

Input: XPath query Q (referring to data values)
Question: does there exists a document,
in which Q is satisfied?

This problem is undecidable!

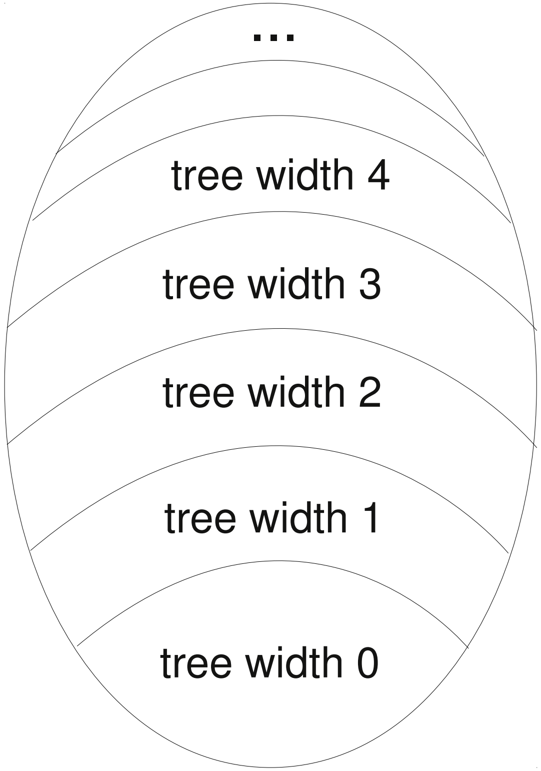
Previous results:
decidable, when query is of special form (Diego Figueira)

Our approach: restrict allowed documents

Analogy: MSO logic on graphs, and tree width

we consider MSO which can quantify over sets of edges

- general satisfiability undecidable

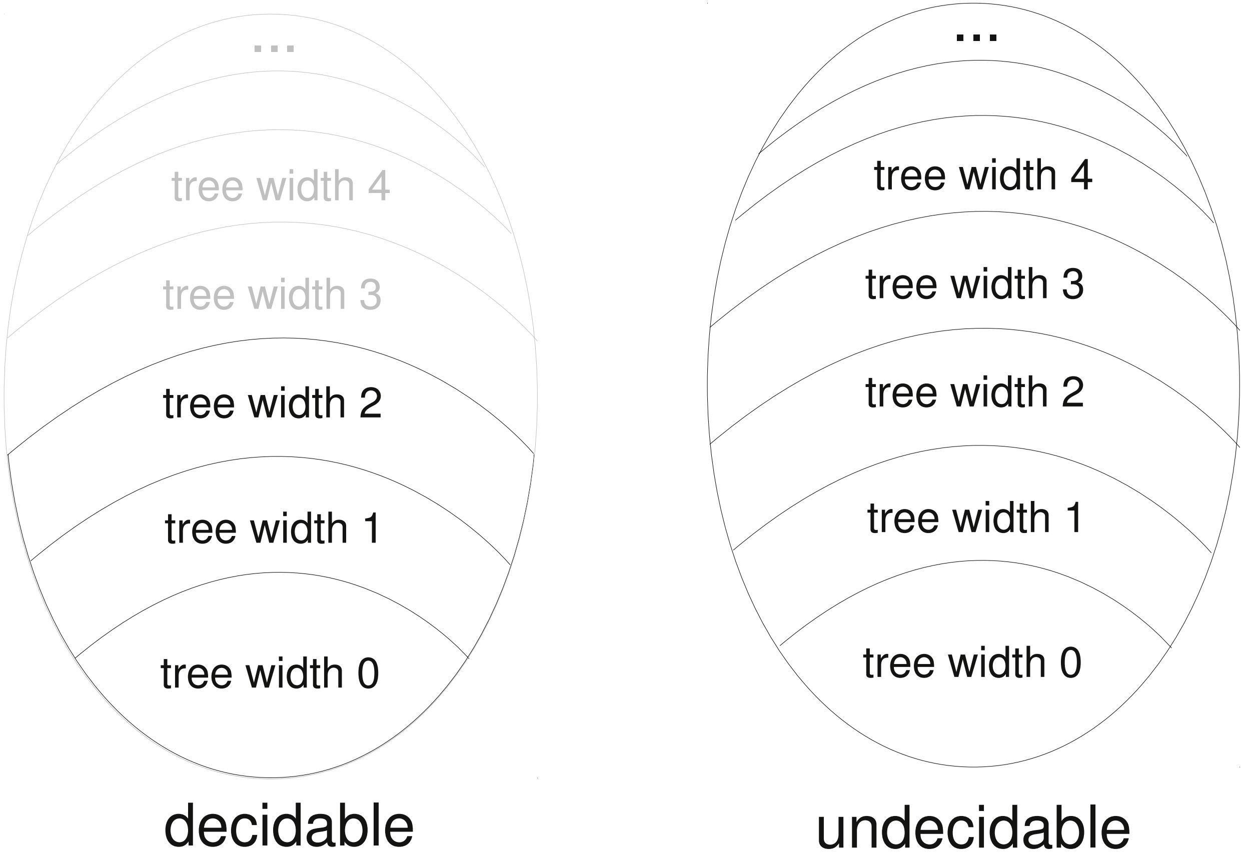


undecidable

Analogy: MSO logic on graphs, and tree width

we consider MSO which can quantify over sets of edges

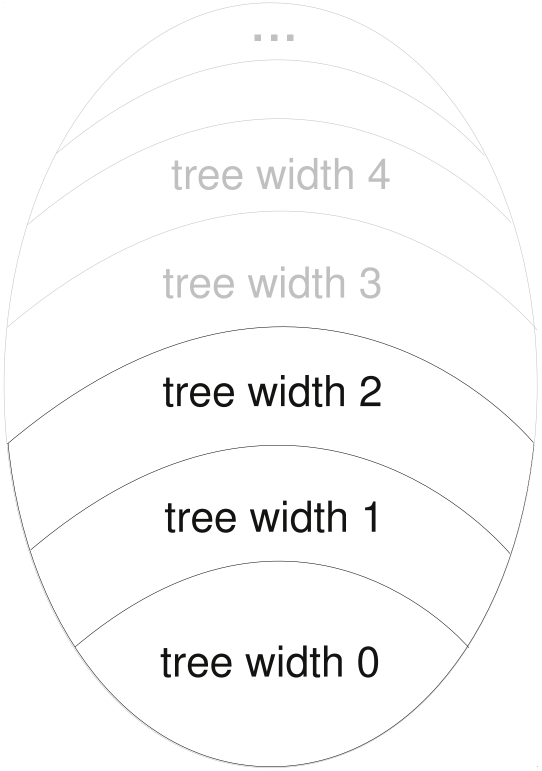
- general satisfiability undecidable
- decidable, when restricted to graphs of tree width $\leq k$



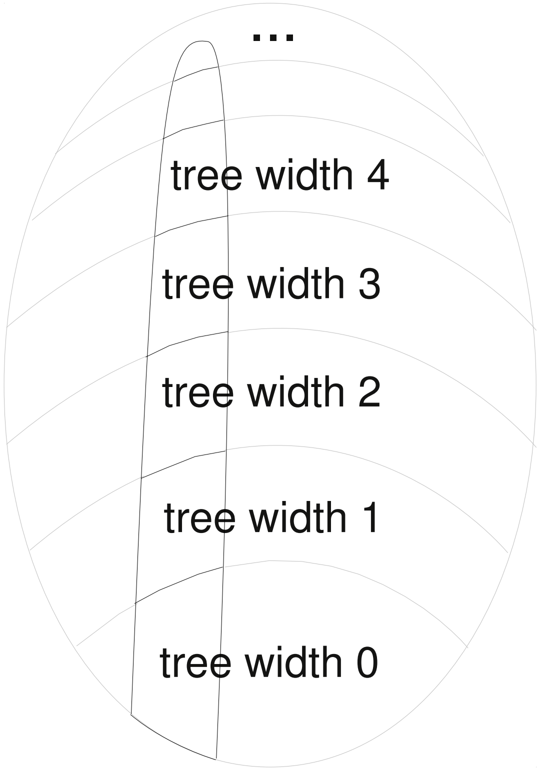
Analogy: MSO logic on graphs, and tree width

we consider MSO which can quantify over sets of edges

- general satisfiability undecidable
- decidable, when restricted to graphs of tree width $\leq k$
- undecidable for classes of graphs with unbounded tree width



decidable

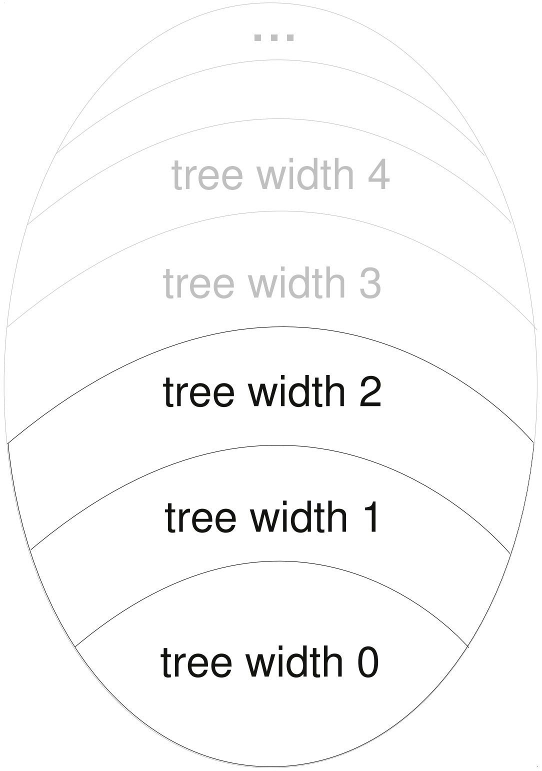


undecidable

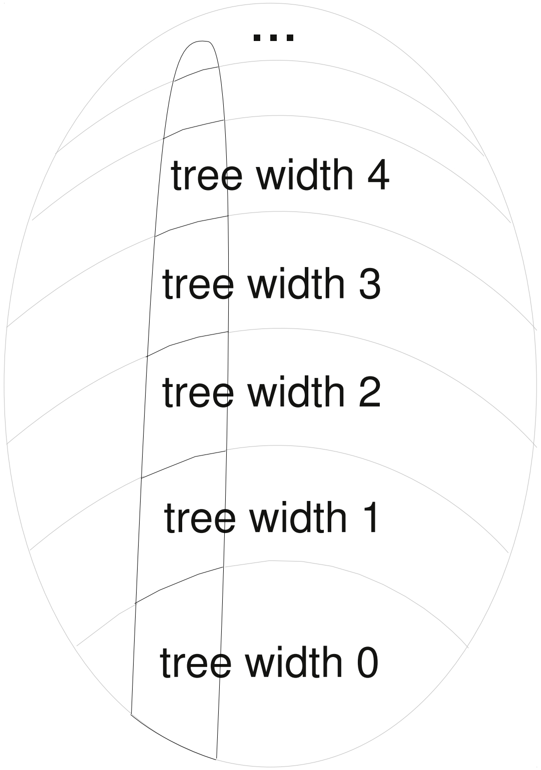
Analogy: MSO logic on graphs, and tree width

we consider MSO which can quantify over sets of edges

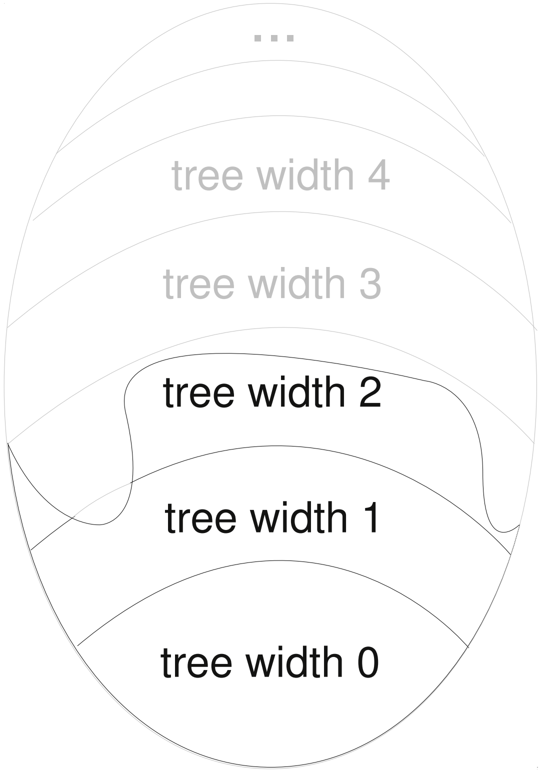
- general satisfiability undecidable
- decidable, when restricted to graphs of tree width $\leq k$
- undecidable for classes of graphs with unbounded tree width



decidable



undecidable



decidable
or
undecidable

There is at most one such measure !!!!

Measure for XML documents (decidability of satisfiability)

Restrictions:

- we consider data words
- every data value appears exactly twice

Data words with the same arrangement of data have the same measure (labels are ignored)

Measure for XML documents (decidability of satisfiability)

Restrictions:

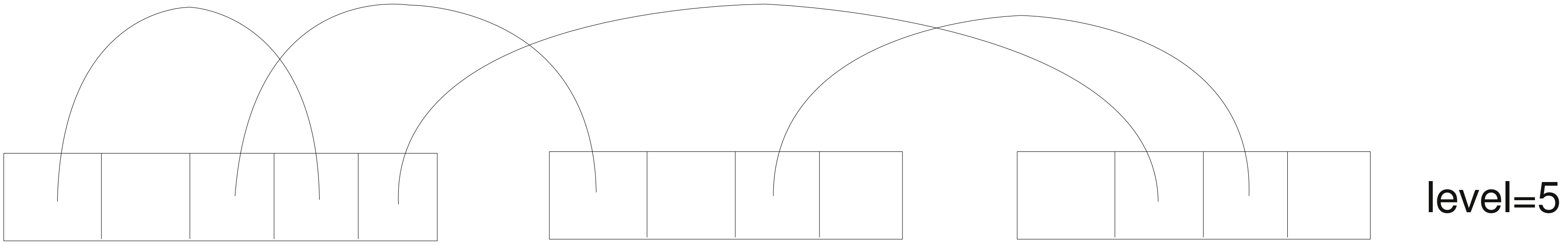
- we consider data words
- every data value appears exactly twice

Data words with the same arrangement of data have the same measure (labels are ignored)

We generate data words using some operations.

Intermediate object - split data word:

- a data word divided into parts (unordered)
- some positions are matched, some are not matched
- it has associated a number, called level



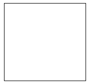
Measure for XML documents (decidability of satisfiability)

Intermediate object - split data word:

- a data word divided into parts (unordered)
- some positions are matched, some are not matched
- it has associated a number, called level

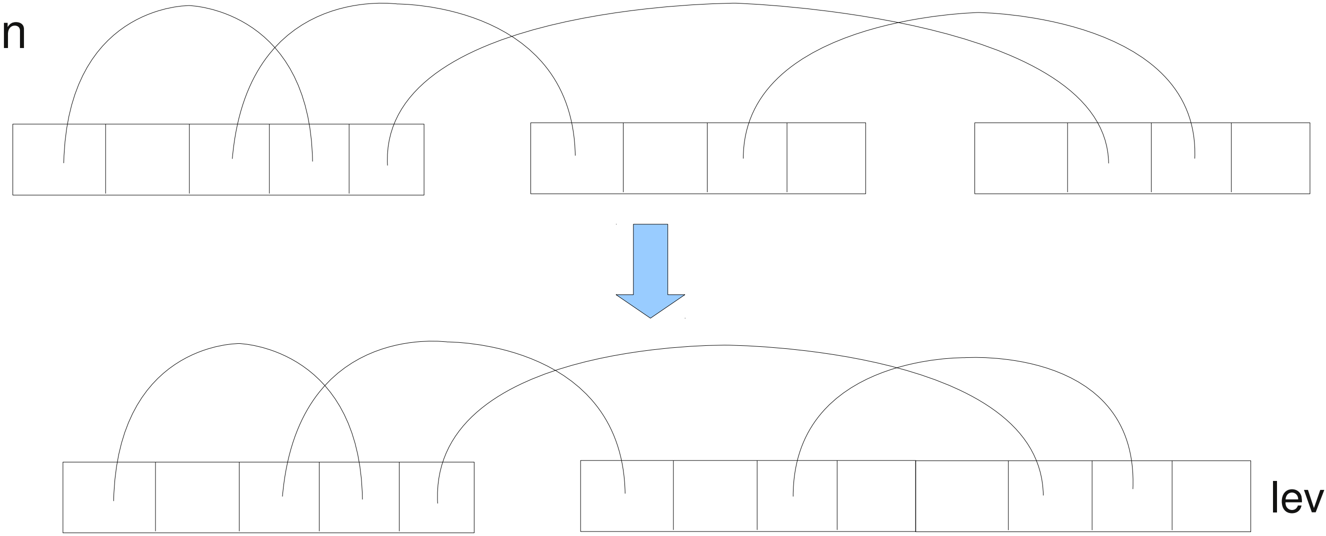
Operations:

- constant



level = 0

- join



level = n

level = n

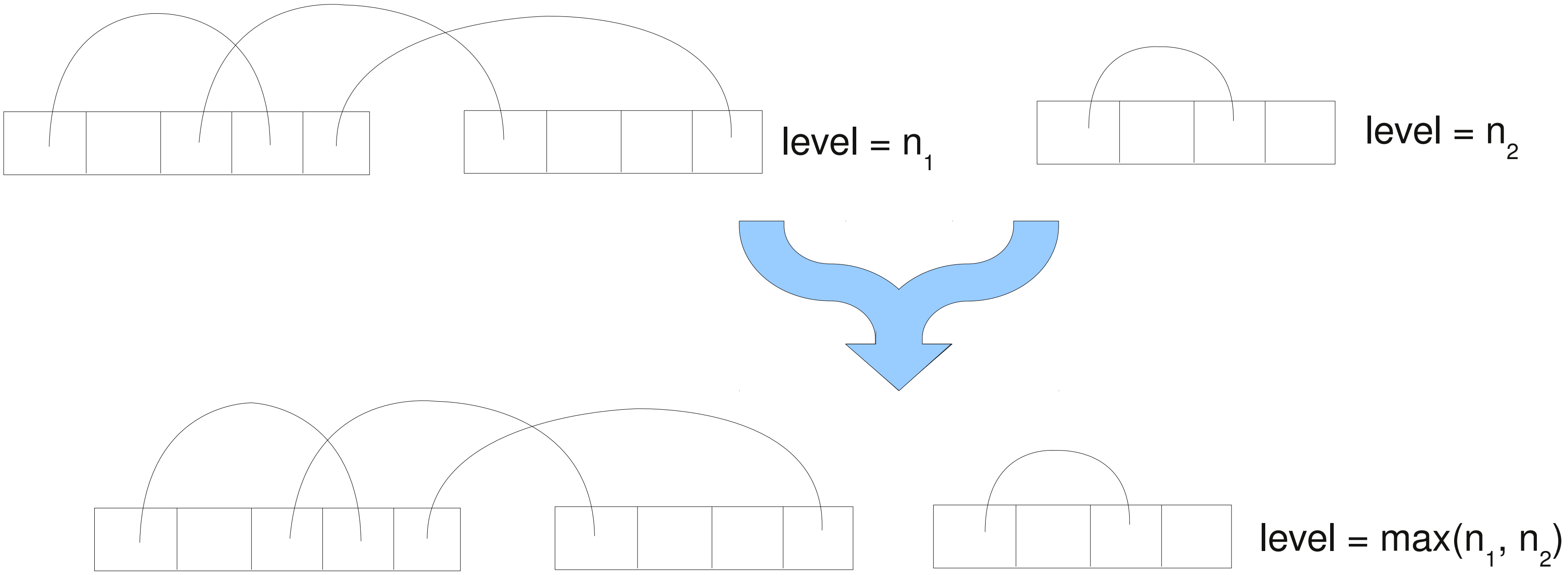
Measure for XML documents (decidability of satisfiability)

Intermediate object - split data word:

- a data word divided into parts (unordered)
- some positions are matched, some are not matched
- it has associated a number, called level

Operations:

- constant
- join
- sum



level = n_1

level = n_2

level = $\max(n_1, n_2)$

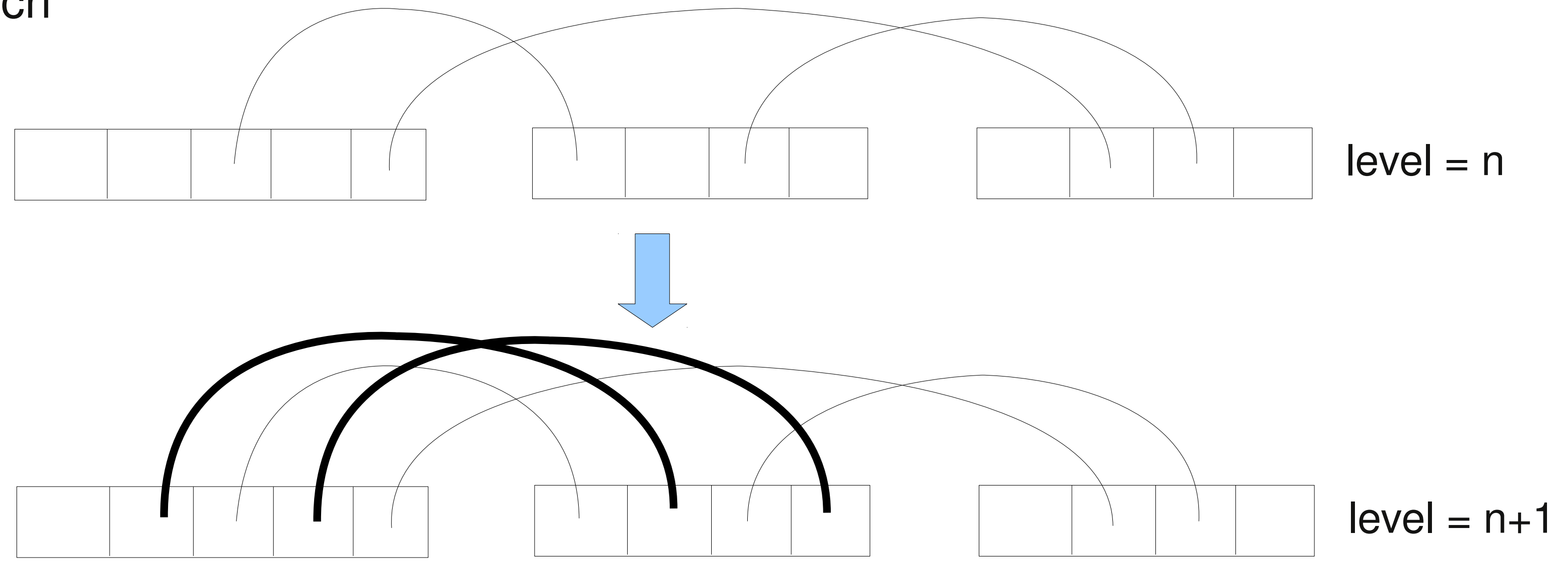
Measure for XML documents (decidability of satisfiability)

Intermediate object - split data word:

- a data word divided into parts (unordered)
- some positions are matched, some are not matched
- it has associated a number, called level

Operations:

- constant, join, sum
- match



Choose some unmatched positions in two different parts, and match them; increase the level

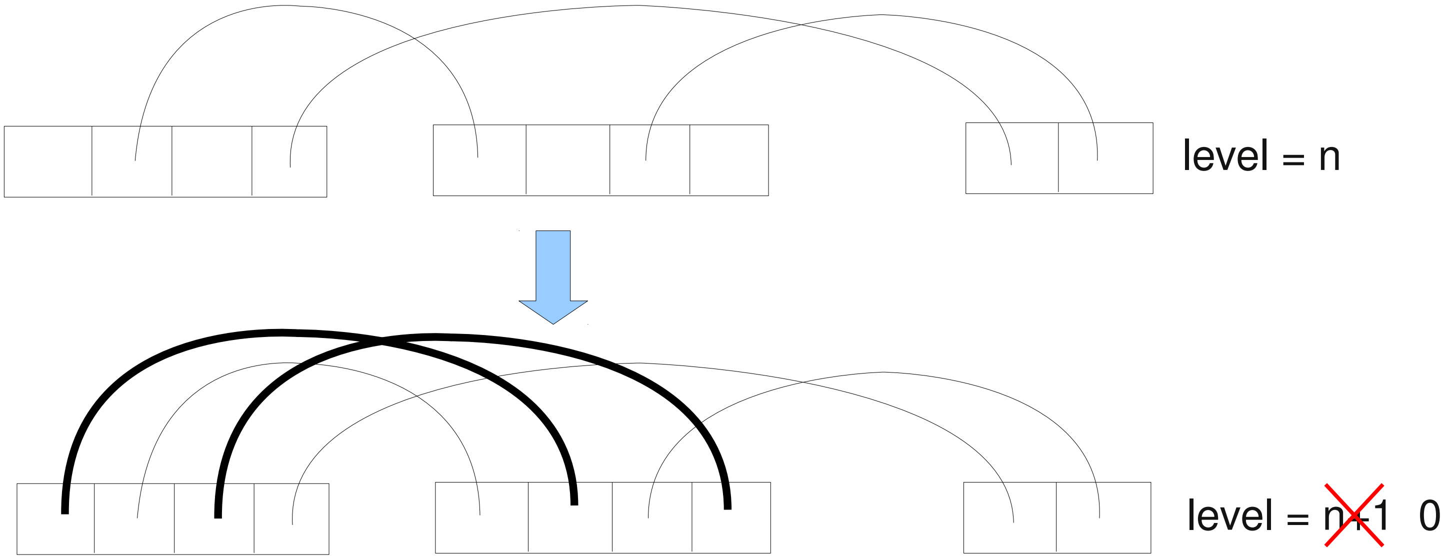
Measure for XML documents (decidability of satisfiability)

Intermediate object - split data word:

- a data word divided into parts (unordered)
- some positions are matched, some are not matched
- it has associated a number, called level

Operations:

- constant, join, sum
- match



level = n

level = ~~n+1~~ 0

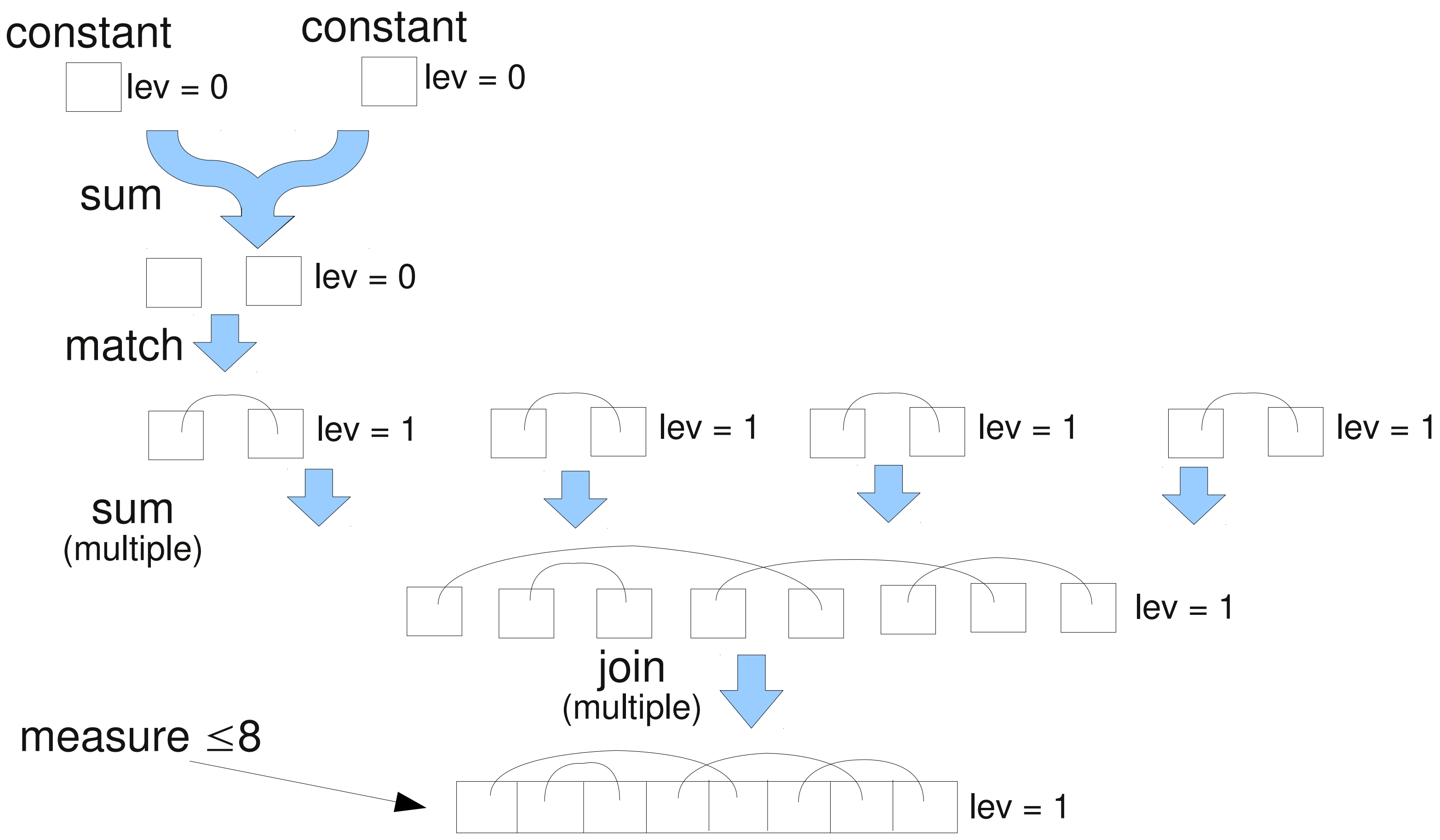
Choose some unmatched positions in two different parts, and match them; increase the level; the level becomes 0 if all positions are matched

Measure for XML documents (decidability of satisfiability)

Definition

If a document can be generated using levels ≤k, and ≤k parts of a split data words then it has complexity measure ≤k.

Example: generate arbitrary data word

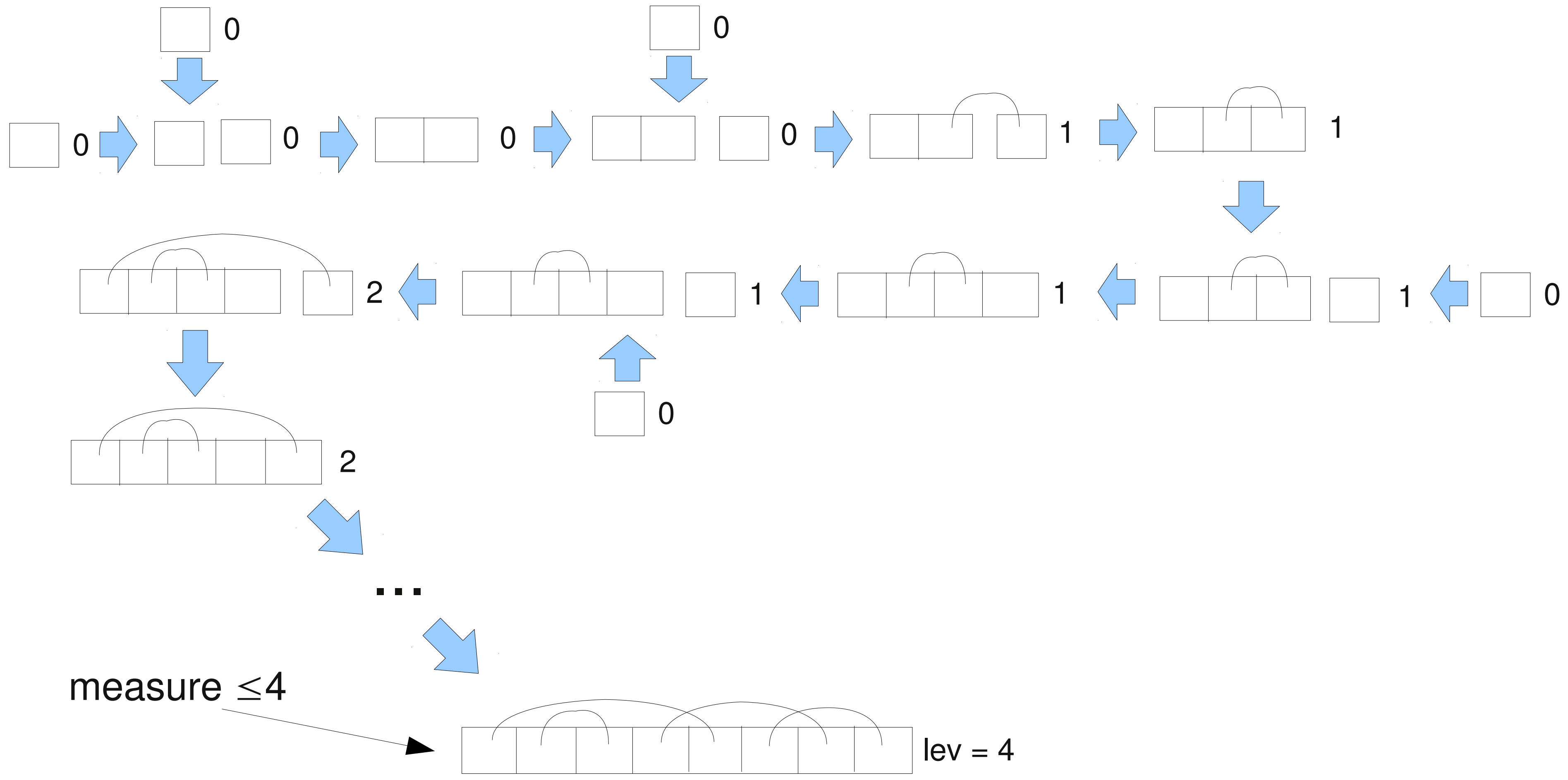


Measure for XML documents (decidability of satisfiability)

Definition

If a document can be generated using levels ≤k, and ≤k parts of a split data words then it has complexity measure ≤k.

Example: generate arbitrary data word (approach 2)



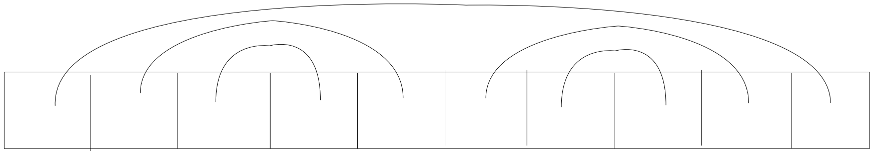
Measure for XML documents (decidability of satisfiability)

Definition

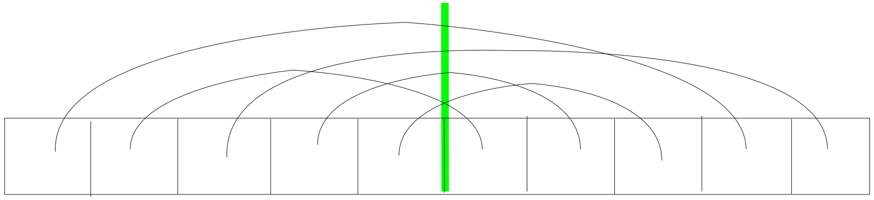
If a document can be generated using levels ≤k, and ≤k parts of a split data words then it has complexity measure ≤k.

Example: words with small measure

well parenthesized words



segregated words



Decidability

<u>Theorem</u>

The following problem is decidable:

Input: XPath query Q (referring to data values),
number k

Question: does there exists a data word of measure $\leq$k,
in which Q is satisfied?

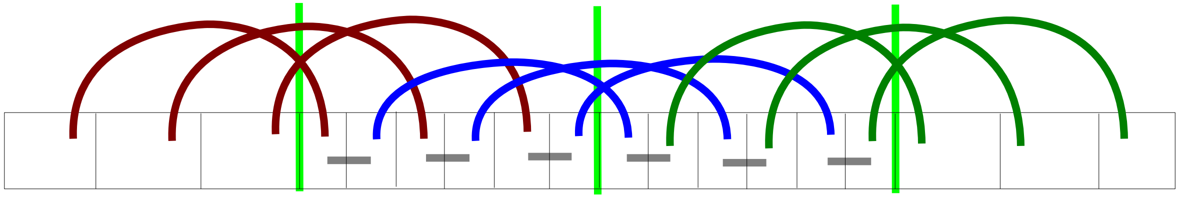
The same works also for Regular XPath, or class automata

Used techniques:

- class automata
- semilinear sets
- Parikh theorem

Undecidability

Example



If X contains arbitrarily large such words,
it is undecidable whether there is a word in X satisfying a given query

Notice: it is not enough to use this grid – the query has to say
that the word is of such form

Undecidability

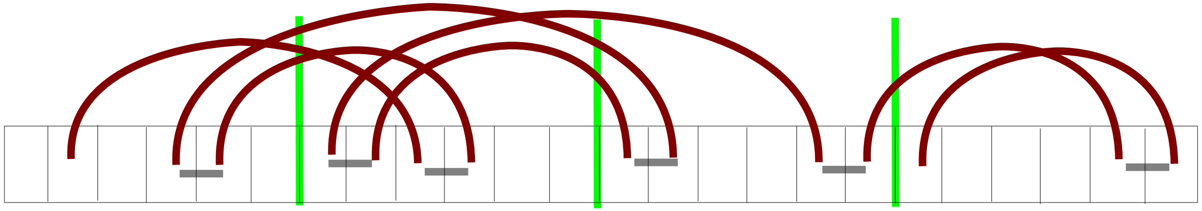
(for undecidability we define another measure;
unfortunately we don't know whether these two measures coincide)

Path measure (analogy: grid as a minor)

measure $\geq$n ⇔ word can be divided into n segments,
so that we can find n disjoint paths

Definition of a path:

- uses data edges and successor edges
- visits every segment
- after visiting segment k does not visit segment k-2



Theorem. If X contains arbitrarily large such words,
it is undecidable whether there is a word in X satisfying a given query

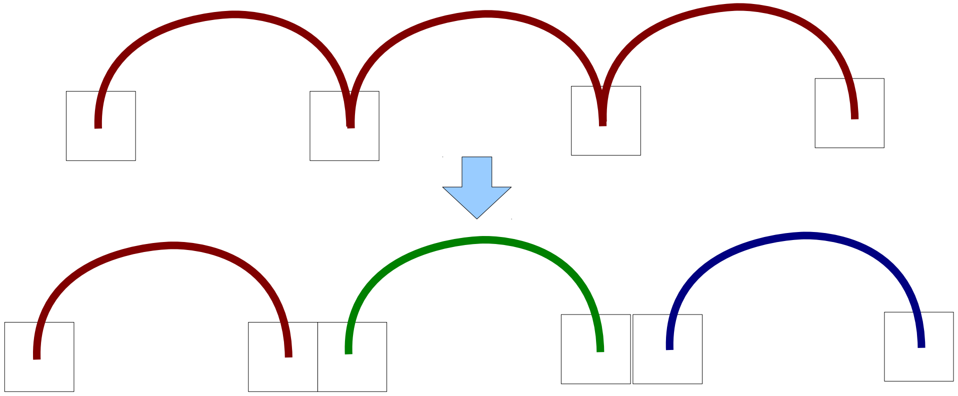
Conclusions

The satisfiability problem is
- decidable in the class of data words with bounded measure
- undecidable in any class containing words with unbounded path measure

Questions:

- are the two measures equivalent?
- words with more than two appearances of a data value

one idea: replace a big class by many classes of size 2:



measure of the original word $\overset{\text{def}}{=}$ measure of the word after translation