CSE 544
Theory of Query Languages
Announcements

• **Project Milestone:** was due on Friday
  – Expect feedback by the end of this week

• **Project presentations:** Friday, 6/7
  – Reserve all day, stay tuned for announcements

• **Homework 3:** was due yesterday
  – Did you remember to turn off your servers!?!?

• **Homework 4:** will be posted in 2-3 days

• **Next paper review:** next Wednesday, 5/29
Complexity Classes

A decision problem:

• We have a property (a.k.a. problem)
• Given an input $X$ of size $n$, decide if it satisfies the property

• In other words, define have to compute a function $f(X) = 0$ or $1$
We are interested in these classes
The Class $\text{AC}^0$

What is $\text{AC}^0$?
The Class $\text{AC}^0$

A problem $f$ is in $\text{AC}^0$ if, for every $n$, there exists a Boolean circuit s.t.:

• It consists of unbounded fan-in AND, unbounded fan-in OR, and NOT gates
• If the inputs $X_1, \ldots, X_n$ encode an input $X$, then the circuit’s output is $f(X)$
• The circuit size is $n^{O(1)}$
• The circuit depth is $O(1)$
Example in AC$^0$

**Problem**: given an input string $X_1, \ldots, X_n$ in $\{0,1\}^n$, check if it has at least two 1’s

0100101 - yes
00001000 - no
Example in $AC^0$

**Problem:** given an input string $X_1, \ldots, X_n$ in $\{0,1\}^n$, check if it has at least two 1’s

0100101 - yes  00001000 - no

Size = $n(n-1)/2+1$
Depth = 2
Are these in $\text{AC}^0$ or not?

**Carry bit:** The sum $(X_nX_{n-1}...X_1) + (Y_nY_{n-1}...Y_1)$ has a carry bit

- $(1001)+(0101) = (1110)$ NO
- $(1001)+(0111) = (10000)$ YES

**Triangle:** A graph given by the $n \times n$ adjacency matrix contains a directed triangle

| 0 1 0 0 0 | YES |
| 0 0 1 0 0 |
| 0 0 0 1 1 |
| 1 0 0 0 0 |
| 0 1 0 0 0 |

**Parity:** $X_1,...,X_n$ has an even number of 1’s

- 10010101101 YES
- 10010101100 NO

**s-t Reachability (GAP):** A graph given by the $n \times n$ adjacency matrix contains a path from node $s$ to node $t$
Are these in $\text{AC}^0$ or not?

**Carry bit:** The sum $(X_nX_{n-1}...X_1) + (Y_nY_{n-1}...Y_1)$ has a carry bit.

In $\text{AC}^0$:

- $(1001)+(0101) = (1110)$ NO
- $(1001)+(0111) = (10000)$ YES

**Triangle:** A graph given by the $n \times n$ adjacency matrix contains a directed triangle.

In $\text{AC}^0$:

| 0 1 0 0 0 |
| 0 0 1 0 0 |
| 0 0 0 1 1 |
| 1 0 0 0 0 |
| 0 1 0 0 0 |

YES

**Parity:** $X_1, ..., X_n$ has an even number of 1's.

Not in $\text{AC}^0$:

- $10010101101$ YES
- $10010101100$ NO

**s-t Reachability (GAP):** A graph given by the $n \times n$ adjacency matrix contains a path from node $s$ to node $t$.

Not in $\text{AC}^0$:

Make sure you understand why!
Theorem: Every Boolean relational query defines a property that is in $\text{AC}^0$.
Example

```
select distinct R.A, S.C
from R, S
where R.B=S.B
```
Prove that Q is in AC$^0$

Example circuit for n = 3 (i.e. ADom={a,b,c})

Q = $\exists z. R('a',z) \land S(z,'c')$

select distinct R.A, S.C from R, S where R.B=S.B
Example

Q = \exists z. R('a', z) \land S(z, 'c')

Example circuit for n = 3 (i.e. ADom={a,b,c})

\textbf{Prove that Q is in AC}^{0}
Example circuit for $n = 3$ (i.e. $\text{ADom} = \{a, b, c\}$)

$Q = \exists z. R(\text{`}a\text{',}z) \land S(z,\text{`}c\text{'}\text{')}$

Prove that $Q$ is in $\text{AC}^0$

```
select distinct R.A, S.C
from R, S
where R.B=S.B
```
Example

Q = ∃z.R(‘a’,z) ∧ S(z,’c’)

Example circuit for n = 3 (i.e. ADom={a,b,c})
Example

Q = \exists z. R('a', z) \land S(z, 'c')

Example circuit for n = 3 (i.e. ADom=\{a,b,c\})
Example circuit for n = 3 (i.e. ADom={a,b,c})

Q = ∃z. R('a',z) ∧ S(z,'c')

Prove that Q is in AC^0

select distinct R.A, S.C from R, S where R.B=S.B

Circuit depth = 2

OR has n inputs

Each AND has 2 inputs

R: | a | b | c |
---|---|---|---|
| a | 1 | 1 | 0 |
| b | 0 | 1 | 0 |
| c | 0 | 0 | 0 |

S: | A | B | C |
---|---|---|---|
| a | a | c |
| b | b | c |
| a | b | b |

0 0 0 0
0 0 1 1
0 0 1 1
Another Example

\[ Q = \exists y. R('a', y) \land (\forall z. S(y, z) \rightarrow \exists u. R(z, u)) \]

Practice at home:
Show that Q is in AC^0 by showing how to construct a circuit for computing Q.
What is the depth?
What fanouts have your OR and AND gates?
Discussion

In class: make sure you understand very well why every relational query is in $AC^0$

• **Consequence 1** (for theoreticians and their friends):
  – SELECT-DISTINCT-FROM-WHERE queries cannot express PARITY, GAP

• **Consequence 2** (for fans of Big Data)
  – “SQL is embarrassingly parallel”
The Classes L and NL
LOGSPACE and NLOGSPACE

• What is LOGSPACE (or L) ?

• What is NLOGSPACE (or NL) ?
LOGSPACE and NLOGSPACE

• A problem is in LOGSPACE (or $L$) if it can be computed by a deterministic Turing machine using $O(\log n)$ space

• A problem is in NLOGSPACE (or $NL$) if it can be computed by a non-deterministic Turing machine using $O(\log n)$ space

$O(\log n)$ space refers to the working tape: the input is on a separate tape of size $n$
Examples

• **GAP** is in **NLOGSPACE** (why?)

• **1-GAP** (each node has outdegree \( \leq 1 \)) is in **LOGSPACE** (why?)

• **Recall** that none of these problems is in **\( \text{AC}^0 \)**

• **Theorem:** **GAP** is complete for **NLOGSPACE**

• **Theorem:** **1-GAP** is complete for **LOGSPACE**
Datalog

It can express GAP in many ways!!

This proves that datalog can express strictly more queries than the relational calculus.
The Classes PTIME, NP, PSPACE

PTIME, NP, PSPACE

AC^0, L, NL, NC^k, ..., PTIME, NP, PSPACE, All computable problems
The Classes PTIME, NP, PSPACE

• What is PTIME?

• What is NP?

• What is PSPACE?
What is the Complexity of Datalog?

All computable problems

PSPACE

NP

PTIME

NC^k ...

NL

L

AC^0
The Same-Generation Problem

**Problem**: We have a database of microbes, where each microbe $x$ may have several children $y$:

$\text{Parent}(x,y)$

Find all microbes in the same generation with “M62251”

Can we solve it in **datalog**?  Is this problem in **NLOGSPACE**?
The Same-Generation Problem

**Problem**: We have a database of microbes, where each microbe \( x \) may have several children \( y \):

\[
\text{Parent}(x,y)
\]

Find all microbes in the same generation with “M62251”

**Can we solve it in datalog?**

**Is this problem in \textbf{NLOGSPACE}?**

**YES:**

\[
\text{SG}(y,z) :- \text{Parent}(x,y), \text{Parent}(x,z) \\
\text{SG}(y,z) :- \text{SG}(u,v), \text{Parent}(u,y), \text{Parent}(v,y) \\
\text{Answer}(z) :- \text{SG}(“M62251”,z)
\]
Problem: We have a database of microbes, where each microbe $x$ may have several children $y$:

$\text{Parent}(x,y)$

Find all microbes in the same generation with “M62251”

Can we solve it in datalog?

YES:

$\text{SG}(y,z) :- \text{Parent}(x,y), \text{Parent}(x,z)$

$\text{SG}(y,z) :- \text{SG}(u,v), \text{Parent}(u,y), \text{Parent}(v,y)$

$\text{Answer}(z) :- \text{SG}(\text{"M62251"},z)$

Is this problem in $\text{NLOGSPACE}$?

YES! (in class…).
Discussion

• The same-generation problem was a trap:
  – SG is no more complex than GAP!
  – Lesson: GAP is more than meets the eyes

• But datalog is more expressive than \textsc{NLOGSPACE}: it captures all of PTIME, in ways we discuss next, in three steps
Step 1: Complexity of Datalog

**Theorem.** Datalog is in PTIME.

More precisely, fix any Boolean datalog program $P$. The problem: given $D$, check if $P(D) = \text{true}$ is in PTIME.

Proof: … [discuss in class]
Step 1: Complexity of Datalog

**Theorem.** Datalog is in PTIME.

More precisely, fix any Boolean datalog program P. The problem: given D, check if \( P(D) = \text{true} \) is in PTIME.

Proof: … [discuss in class]

Which of the following are in PTIME? Stratified, inflationary-fixpoint, partial-fixpoint datalog?.
Step 2: The Circuit Value Problem

**Input** = a rooted DAG; leaves labeled 0/1, internal nodes labeled AND/OR

**Output** = check if the value of the root is 1

Note: NOT nodes could be added w.l.o.g. (why?)
Step 2: The Circuit Value Problem

**Input** = a rooted DAG; leaves labeled 0/1, internal nodes labeled AND/OR
**Output** = check if the value of the root is 1

Note: NOT nodes could be added w.l.o.g. (why?)

**Theorem.**
The Circuit Value Problem is complete for PTIME

In class:
1. How can we compute it in PTIME?
2. Why isn’t it in NLOGSPACE?
Step 2: The Circuit Value Problem

**Theorem.** Datalog can express the Circuit Value Problem

EDBs:
- root(x)
- and(x,y1,y2)
- or(x,y1,y2)
- zeroLeaf(x)
- oneLeaf(x)
Step 2: The Circuit Value Problem

**Theorem.** Datalog can express the Circuit Value Problem

EDBs:

- root(x)
- and(x, y1, y2)
- or(x, y1, y2)
- zeroLeaf(x)
- oneLeaf(x)

oneNode(x) :- oneLeaf(x)
oneNode(x) :- or(x, y1, y2), oneNode(y1)
oneNode(x) :- or(x, y1, y2), oneNode(y2)
oneNode(x) :- and(x, y1, y2), oneNode(y1), oneNode(y2)
Answer() :- root(x), oneNode(x)
Discussion

• Step 1: datalog is in PTIME
  – Stratified, and inflationary datalog\(^\sim\) are in PTIME

• Step 2: datalog can express a PTIME complete problem

• Step 3: can datalog express \textit{all} PTIME problems?
Step 3

**Theorem.** For every problem in PTIME there exists a program in inflationary-fixpoint datalog\(^\neg\) that expresses that problem.

Caveat: the program must have access to a total order on the active domain. Otherwise inflationary-fixpoint datalog\(^\neg\) cannot even express parity!

Make sure you understand why pure datalog (without negation) cannot express all of PTIME.
Finally: partial-fixpoint Datalog$\neg$

**Theorem.** Partial-fixpoint datalog$\neg$ can express precisely the problems that are in PSPACE

Same caveat: for completeness we need access to an order relation
Which are “Easy” to Parallelize?

• Relational calculus = $\text{AC}^0$

• Add transitive closure = $\text{NLOGSPACE}$

• Inflationary datalog = $\text{PTIME}$
Which are “Easy” to Parallelize?

• Relational calculus = $\text{AC}^0$
• YES!! “embarrassingly parallel”

• Add transitive closure = $\text{NLOGSPACE}$
• MAYBE: the path-doubling program

• Inflationary datalog = $\text{PTIME}$
• NO: circuit value problem
Descriptive Complexity

- In **computational complexity** one describes complexity classes in terms of a computational model
  - Turing Machine, circuit, etc
- In **descriptive complexity** one describes complexity classes in terms of the logic ("query language") that captures that class
Descriptive Complexity

[Immerman, Vardi]
Assume we have access to an order relation < (and to a BIT relation for $AC^0$)