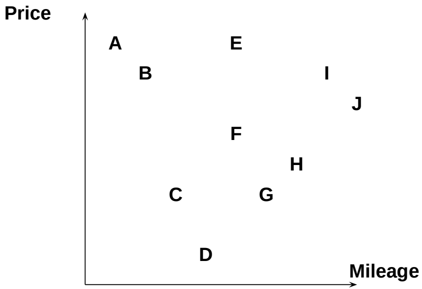
ZINC: Efficient Indexing for Skyline Computation

Bin Liu   Chee-Yong Chan
Department of Computer Science
National University of Singapore

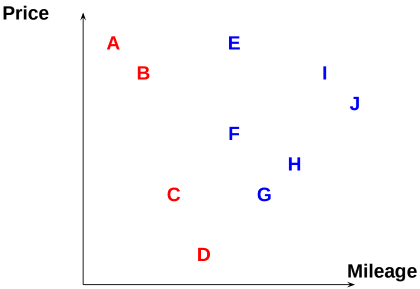
Skyline Queries

- **Skyline** – points that are not dominated by other points wrt a set of dimensions
- Point x **dominates** point y if
  (1) x is as good as y in all dimensions, and
  (2) x is better than y in at least one dimension
- **Example**: Find used cars that are cheap and have low mileage



Skyline Queries

- **Skyline** – points that are not dominated by other points wrt a set of dimensions
- Point x **dominates** point y if
  (1) x is as good as y in all dimensions, and
  (2) x is better than y in at least one dimension
- **Example**: Find used cars that are cheap and have low mileage



Simple Evaluation Algorithm

Input: set of data points *P*
Output: set of skyline points in *P*

initialize set of candidate skyline points *S* to be empty
for each data point *p* in *P* do
    if (*p* is not dominated by any point in *S*) then
        delete each $s \in S$ if *p* dominates *s*
        insert *p* into *S*
return *S*

Simple Evaluation Algorithm

Input: set of data points *P*
Output: set of skyline points in *P*

```
initialize set of candidate skyline points S to be empty
for each data point p in P do
    if (p is not dominated by any point in S) then
      delete each s ∈ S if p dominates s
      insert p into S
return S
```

Drawbacks:

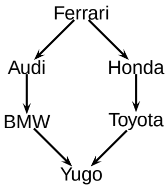
- Need to scan entire data set
- Performs many dominance comparisons
- Non-progressive

Processing Skyline Queries

- **Scan-based solutions:**
  - **BNL, D&C** [Börzsönyi, Kossmann, Stocker, ICDE'01]
  - **SFS** [Chomicki, Godfrey, Gryz, Liang, ICDE'03]
  - **LESS** [Godfrey, Shipley, Gryz, VLDB'05]
  - **LS** [Morse, Patel, Jagadish, VLDB'07]
- **Index-based solutions:**
  - **Bitmap, Index** [Tan, Eng, Ooi, VLDB'01]
  - **NN** [Kossmann, Ramsak, Rost, VLDB'02]
  - **BBS** [Papadias, Tao, Fu, Seeger, SIGMOD'03]
  - **ZB-tree** [Lee, Zheng, Li, Lee, VLDB'07]
  - **OPS, LCRS** [Zhang, Mamoulis, Cheung, SIGMOD'09]
  - **BSkyTree** [Lee, Hwang, EDBT'10]

Partially-Ordered Domains

- Many data have partially-ordered domains:
  - User preferences



  - Interval data (e.g., availability period, price range)
  - Type/class hierarchies (e.g., categorical data)
  - Set-valued domains (e.g., skill sets, hotel facilities)

Our Work: ZINC

- Index method for skyline queries with PO domains
- Inspired by ZB-tree
- **ZB-tree** [Lee, Zheng, Li, Lee, VLDB'07]
  - Index method for totally-ordered domains
  - Outperforms BBS [Papadias, Tao, Fu, Seeger, SIGMOD'03]

Our Work: ZINC

- Index method for skyline queries with PO domains
- Inspired by ZB-tree
- **ZB-tree** [Lee, Zheng, Li, Lee, VLDB'07]
  - Index method for totally-ordered domains
  - Outperforms BBS [Papadias, Tao, Fu, Seeger, SIGMOD'03]
- Related work
  - **SDC**$^{+}$ [Chan, Eng, Tan, SIGMOD'05]
  - **TSS** [Sacharidis, Papadopoulos, Papadias, ICDE'09]

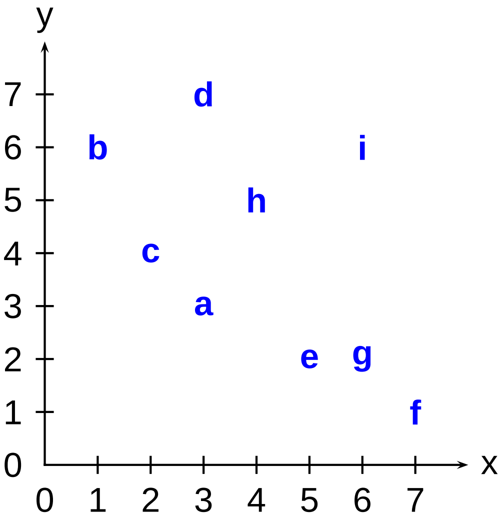
Our Work: ZINC

- Index method for skyline queries with PO domains
- Inspired by ZB-tree
- **ZB-tree** [Lee, Zheng, Li, Lee, VLDB'07]
  - Index method for totally-ordered domains
  - Outperforms BBS [Papadias, Tao, Fu, Seeger, SIGMOD'03]
- Related work
  - **SDC**$^{+}$ [Chan, Eng, Tan, SIGMOD'05]
  - **TSS** [Sacharidis, Papadopoulos, Papadias, ICDE'09]
  - Recent technique:
    - **CPS, SCL** [Zhang, Mamoulis, Cheung, Kao, VLDB'10]

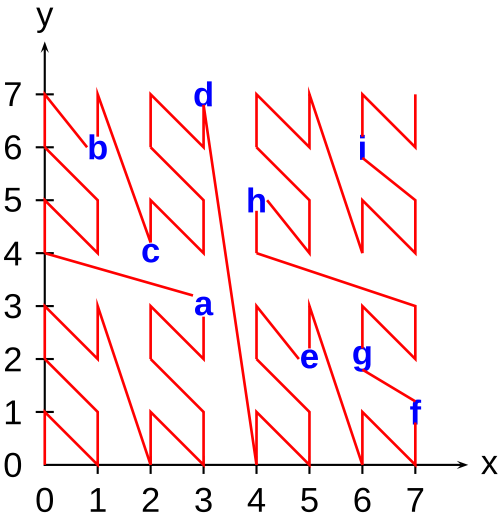
ZB-tree

- Maps multi-dimensional data point to 1-dimensional Z-address
  - Z-address = Interleaved bitstring representation of attribute values
  - Example: (0,5) = (000,101) $\rightarrow$ 010001
- Index Z-addresses using B^+-tree

ZB-tree: Example

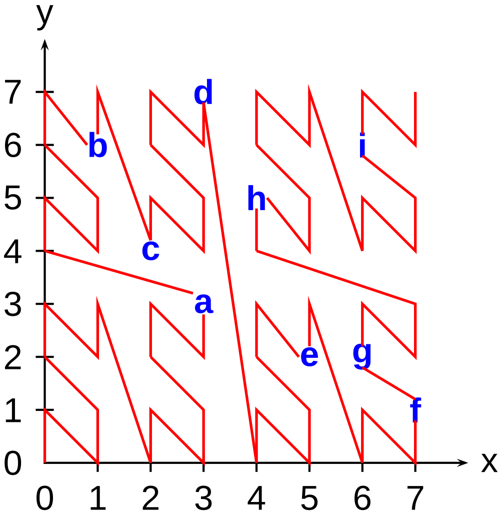


ZB-tree: Example

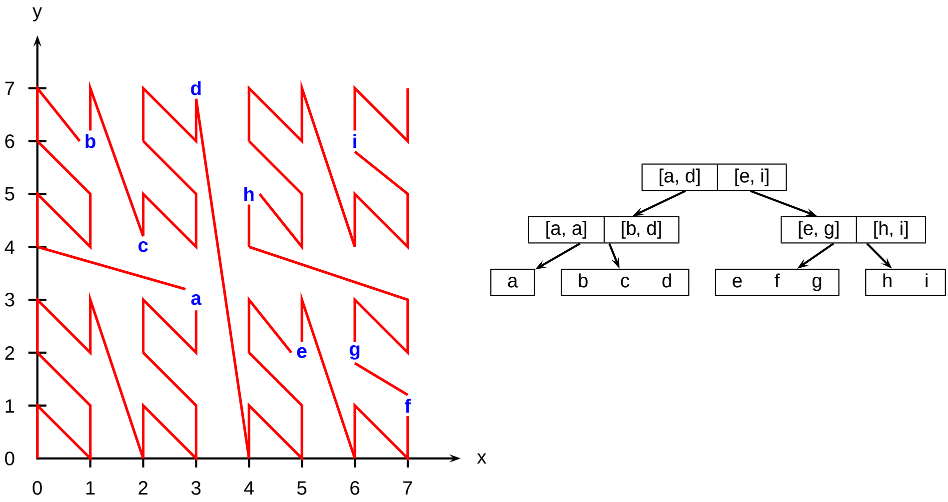


ZB-tree: Example

Monotonic ordering property: if p dominates q, then p precedes q in Z-order



ZB-tree: Example



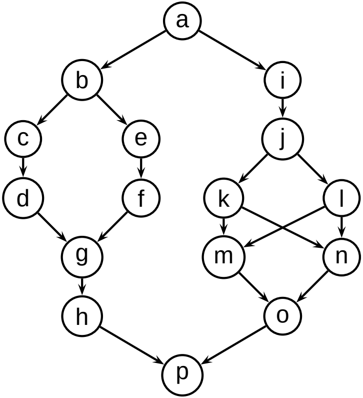
Encoding Schemes for Partial Orders

- Given a partial order domain D, find the smallest set S and an embedding $f : D \to 2^S$ such that x dominates y iff $f(x) \subseteq f(y)$
- Many proposed heuristics:
  - Ait-Kaci et al, ACM TOPLS 1989
  - Caseau, OOPSLA 1993
  - Krall, Vitek, Horspool, ECOOP 1997
  - etc

ZINC: Nested Encoding Scheme

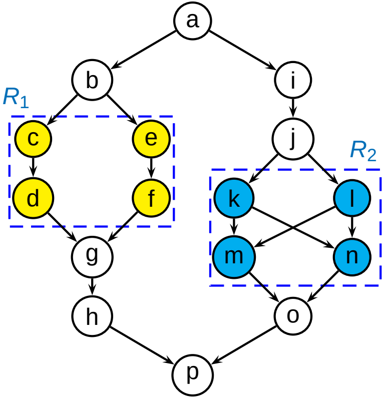
- ZINC = Z-order Indexing with Nested Code
- **Key idea**:
  - Organize PO into nested layers of simpler POs
  - Encode each value in PO as a concatenation of encodings in simpler POs

Example of Partial Order Reduction



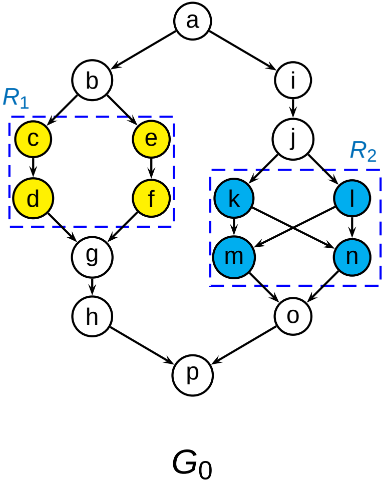
G_0

Example of Partial Order Reduction



G_0

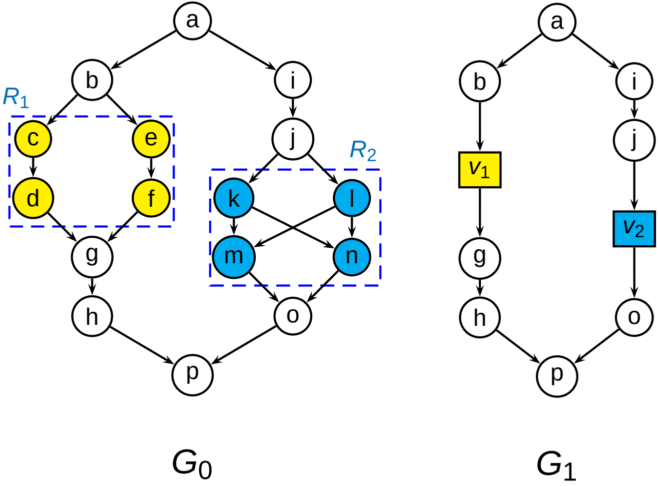
Example of Partial Order Reduction



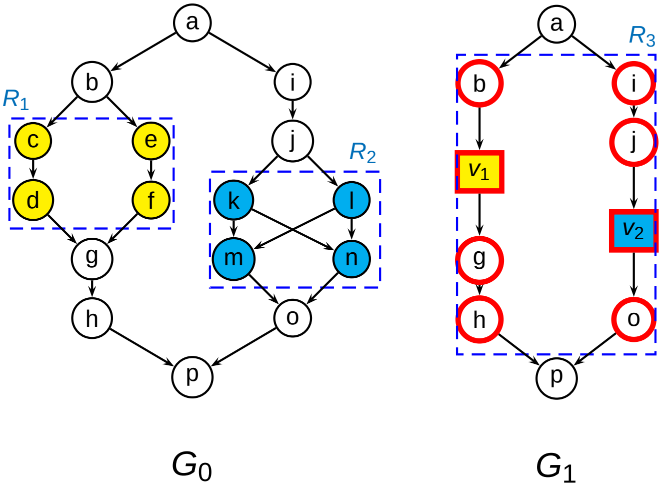
A subset of nodes R in PO is a region if every node in R has the same dominance relationship wrt nodes outside of R

- if $u \in R$ dominates $v \notin R$, then every $u' \in R$ dominates v
- if $v \notin R$ dominates $u \in R$, then v dominates every $u' \in R$

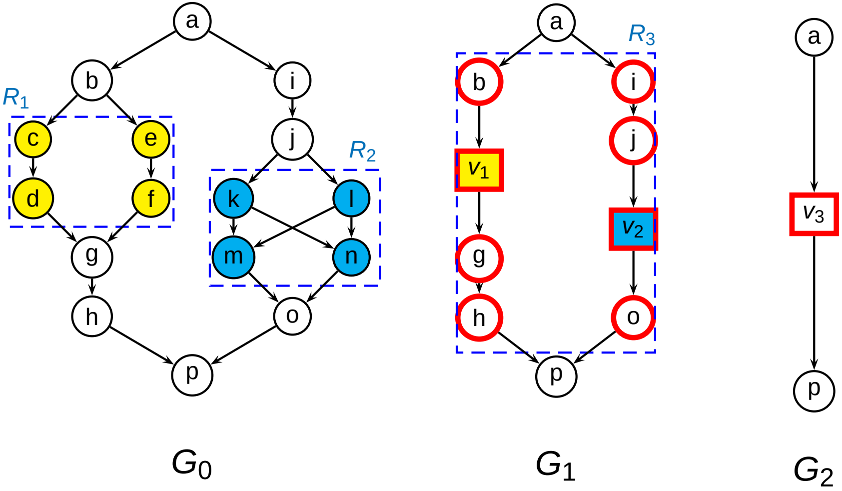
Example of Partial Order Reduction



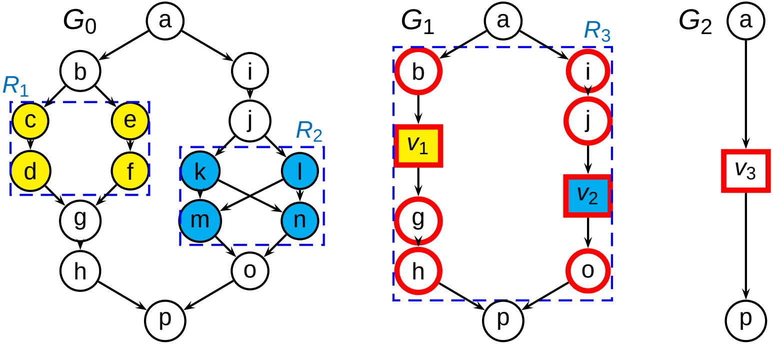
Example of Partial Order Reduction



Example of Partial Order Reduction



Example of Nested Encodings



$\text{Encode}(a, G_0) = \text{Encode}(a, G_2)$

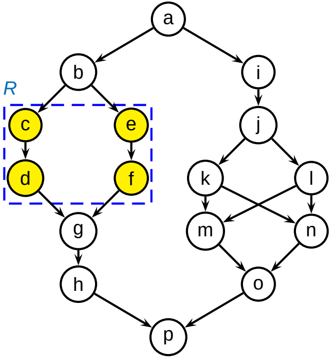
$\text{Encode}(h, G_0) = \text{Encode}(v_3, G_2) + \text{Encode}(h, R_3)$

$\text{Encode}(k, G_0) = \text{Encode}(v_3, G_2) + \text{Encode}(v_2, R_3) + \text{Encode}(k, R_2)$

Vertical Regions

A region R in a PO a vertical region if

- $R = S_0 \cup \cdots \cup S_k$, $k \geq 1$, each S_i is a total order,
- nodes from different total orders are incomparable
- R is maximal subgraph of PO that satisfies the above properties



$R = S_0 \cup S_1$
$S_0 = \{c, d\}$, $S_1 = \{e, f\}$

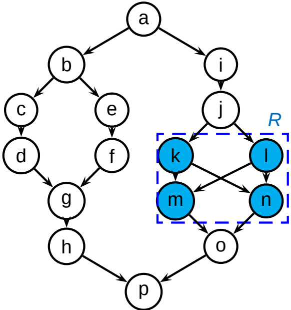
Each $v \in R$ is encoded by two components: (1) which S_i contains v, and (2) rank of v within S_i

$c = 00$, $d = 01$, $e = 10$, $f = 11$

Horizontal Regions

A region R in a PO is a horizontal region if

- $R = S_0 \cup \cdots \cup S_k, k \geq 1$,
- the nodes within each S_i are incomparable,
- $u \in S_i$ dominates $v \in S_j$ if $i < j$, and
- R is maximal subgraph of PO that satisfies the above properties



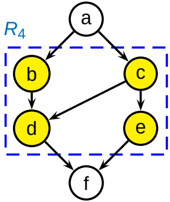
$R = S_0 \cup S_1$
$S_0 = \{k, l\}, \; S_1 = \{m, n\}$

Each $v \in R$ is encoded by i if $v \in S_i$

$k = 0, \; l = 0, \; m = 1, \; n = 1$

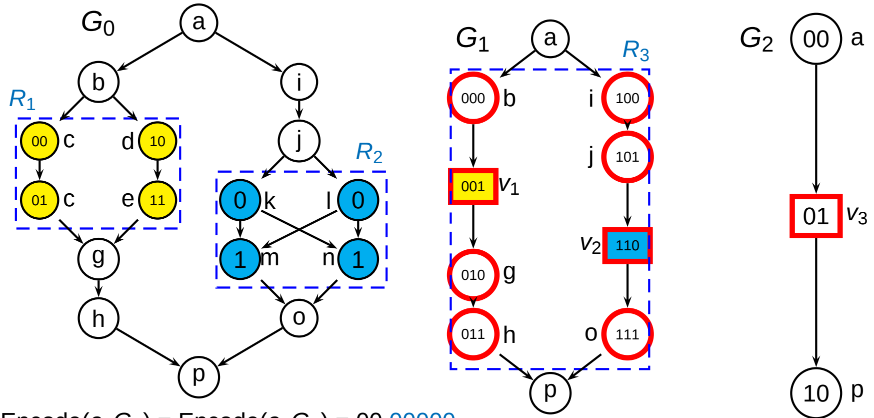
Regular & Irregular Regions

- A region *R* in a PO is a regular region if *R* is either a vertical or horizontal region
- A region *R* in a PO is an irregular region if
  - *R* is not a regular region, and
  - *R* is a minimal subgraph of PO containing at least two nodes
- Example of an irregular region:



- Irregular regions are encoded using Compact Hierarchical Encoding (CHE) [Caseau, OOPSLA 1993]

Putting everything together

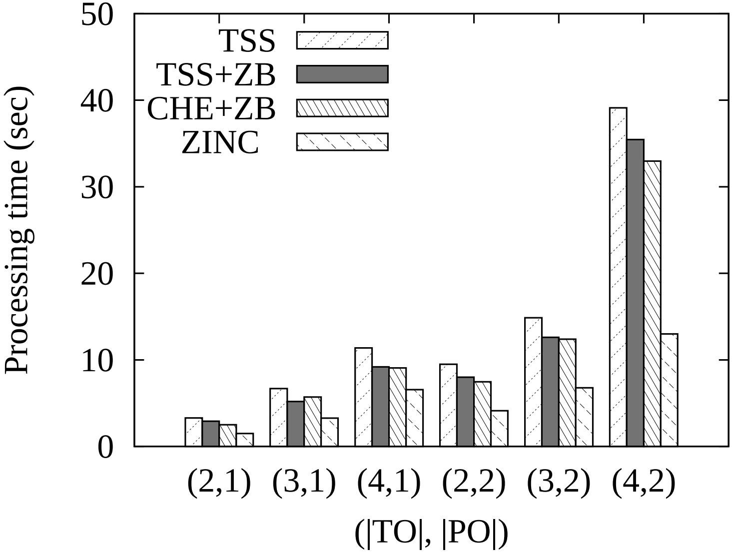


Encode(a, G_0) = Encode(a, G_2) = 00 00000

Encode(h, G_0) = Encode(v_3, G_2) + Encode(h, R_3) = 01 011 00

Encode(k, G_0) = Encode(v_3, G_2) + Encode(v_2, R_3) + Encode(k, R_2) = 01 110 0 0

Performance Comparison



Conclusion

- Presented a novel index method for computing skyline queries on data with partially-ordered attribute domains
- ZINC = Z-order based indexing (ZB-tree) + Nested encoding scheme
- Future work:
  - ZINC vs CPS, SCL [Zhang, Mamoulis, Cheung, Kao, VLDB'10]
  - Other techniques?