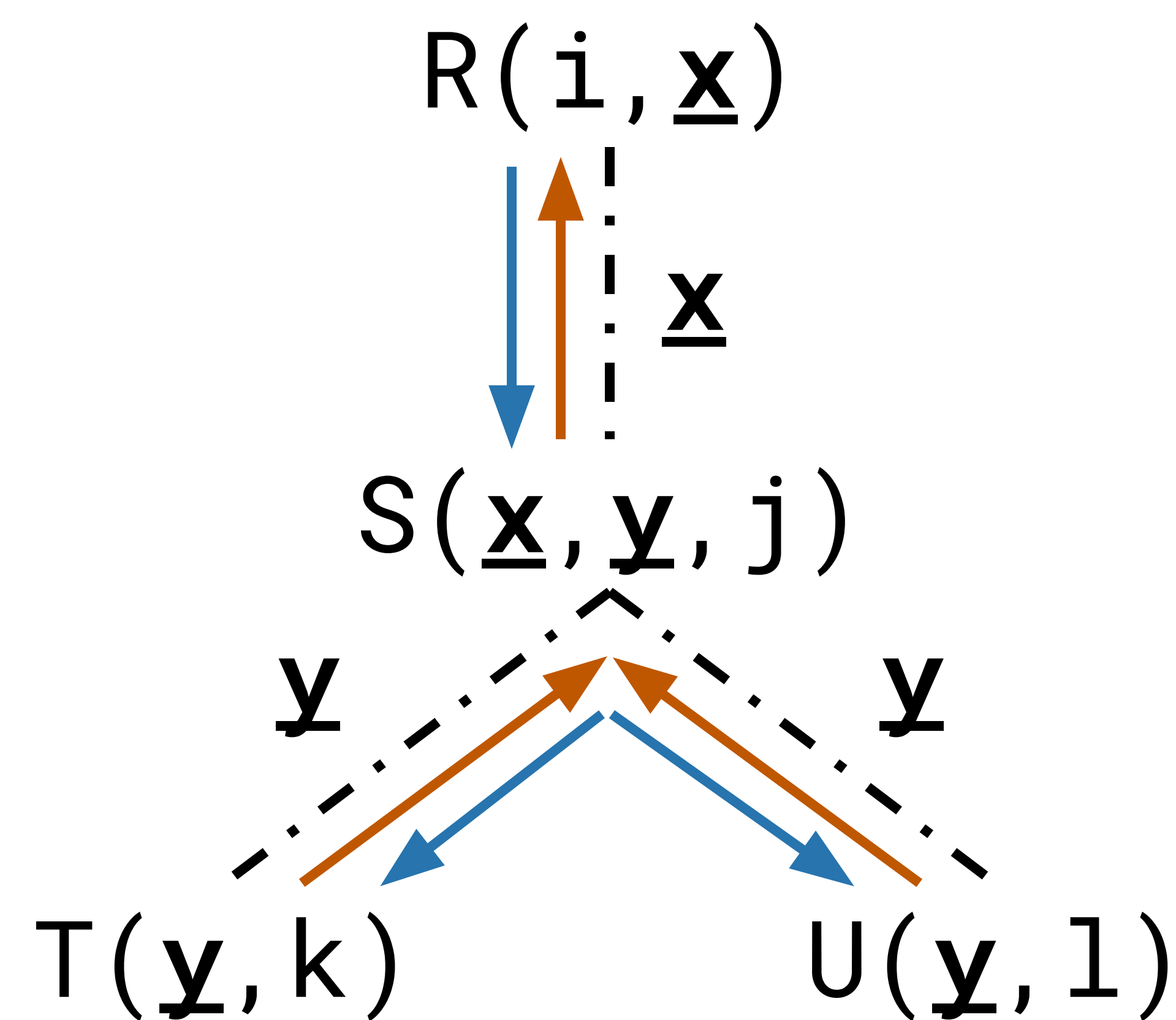


TreeTracker Join: Simple, Optimal, Fast

Zeyuan Hu, Remy Wang, Daniel Miranker

Yannakakis Algo.

```
SELECT *
FROM R, S, T, U
WHERE R.x = S.x
AND S.y = T.y
AND S.y = U.y
```



$S' = S \times T$
 $S'' = S' \times U$
 $R^* = R \times S''$
 $S^* = S'' \times R^*$
 $T^* = T \times S^*$
 $U^* = U \times S^*$

Pipelined Hash Join

```
for i, x in R:
    for y, j in S[x]:
        for k in T[y]:
            for l in u[y]:
                print(...)
```

i	x	x	y	j	y	k	y	l
1	1	1	1	1	1	1	0	1
2	1	1	1	2	1	2	0	2
3	1	1	1	3	1	3	0	3

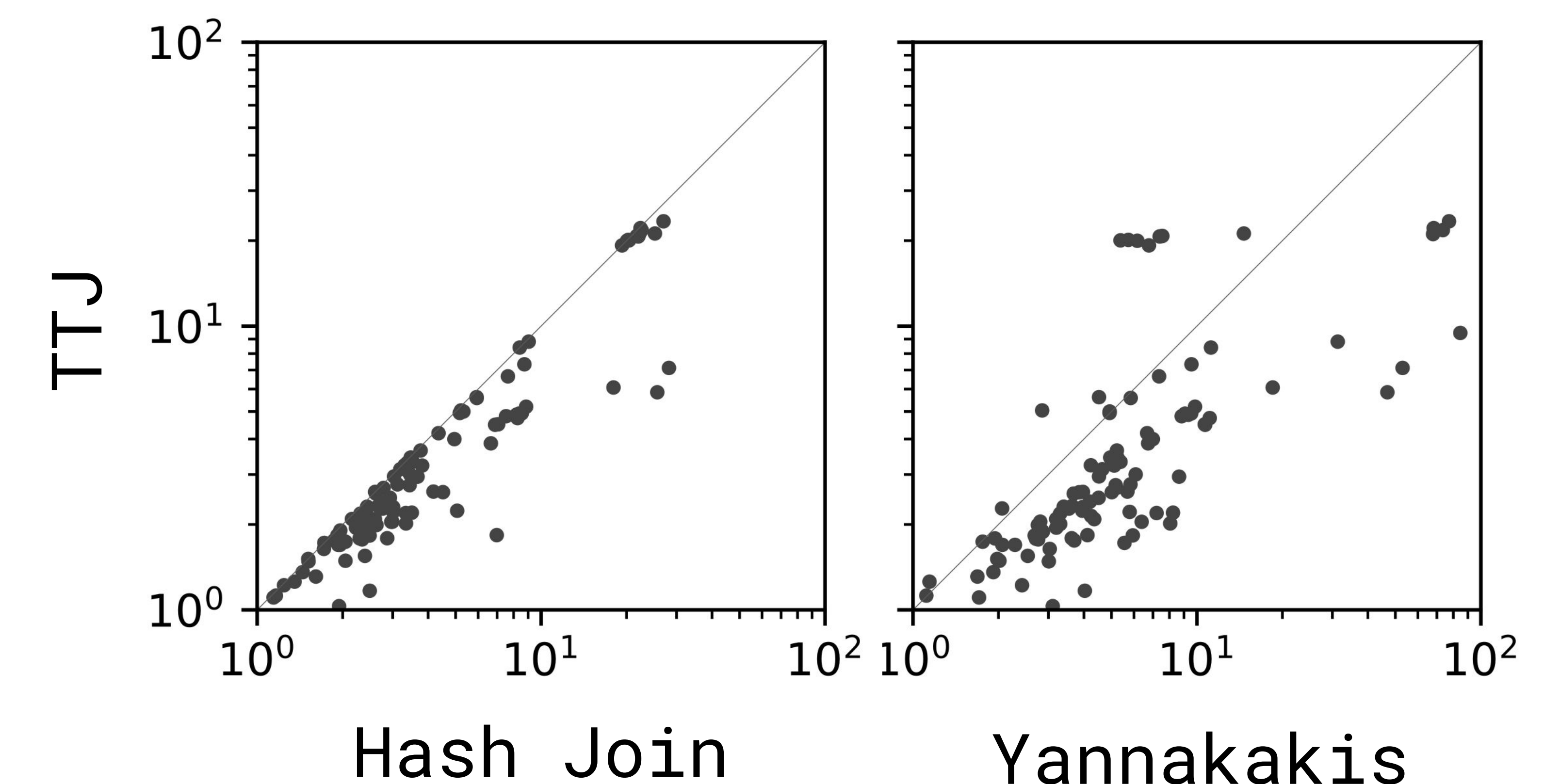
TreeTracker Join

```
for i, x in R:
    for y, j in S[x]:
        for k in T[y]:
            if U[y] is None:
                S[x].del(y, j);
                break
            for l in u[y]:
                print(...)
```

lookup fails: backjump to parent
delete parent row

i	x	x	y	j	y	k	y	l
1	1	1	1	1	1	1	0	1
2	1	1	1	2	1	2	0	2
3	1	1	1	3	1	3	0	3

JOB run time (s)



runs in $O(|IN| + |OUT|)$
 every "run" of TTJ either:
 - outputs ($\leq |OUT|$)
 - deletes ($\leq |IN|$)

#lookup(TTJ) \leq #lookup(HJ)