Comparator automata for quantitative verification*

BY

Suguman Bansal

Swarat Chaudhuri

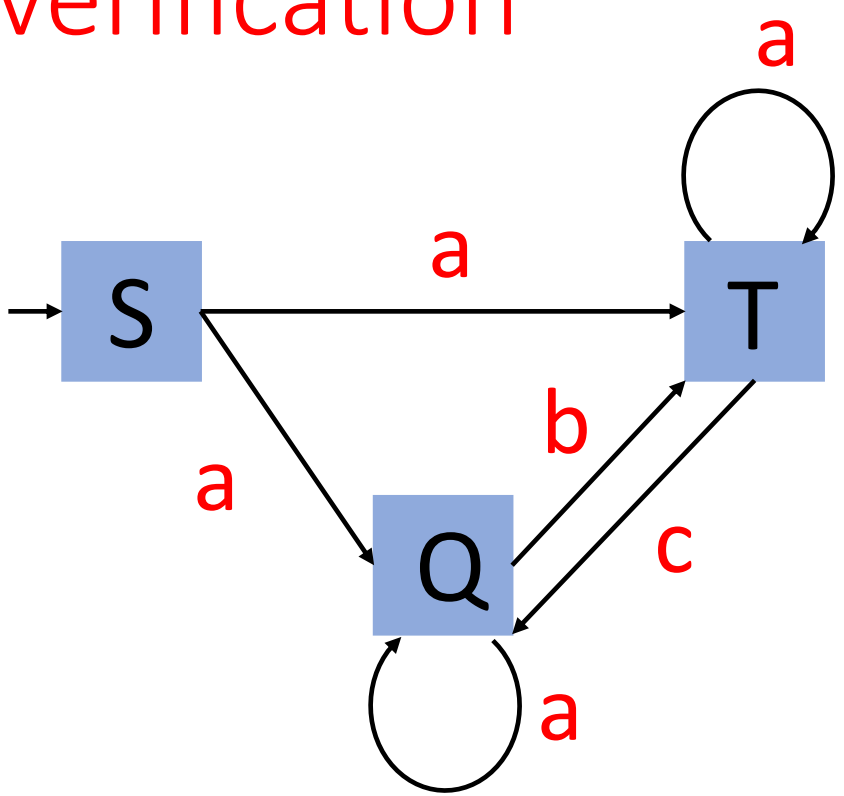
Moshe Y. Vardi

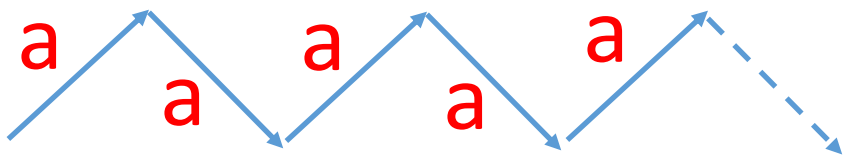
LAV Seminar, UC Berkeley

*Appeared at FoSSaCS 2018 at ETAPS 2018

Comparison is ubiquitous in verification

- Verification is whether a model satisfies a specification
- Weight of word
  - 1, if word present in model or specification
  - 0, otherwise
- Verification
  - Weight of every word in model **is less than or equal to** its weight in specification





Quantitative (quant.) properties

- Real-valued weight of runs
  - Resource consumption of a system
  - Reward received in a strategy

- Quant. verification reasons about quant. properties
  - Does the model consume fewer resource than the specification?
  - Which strategy results in greater reward?

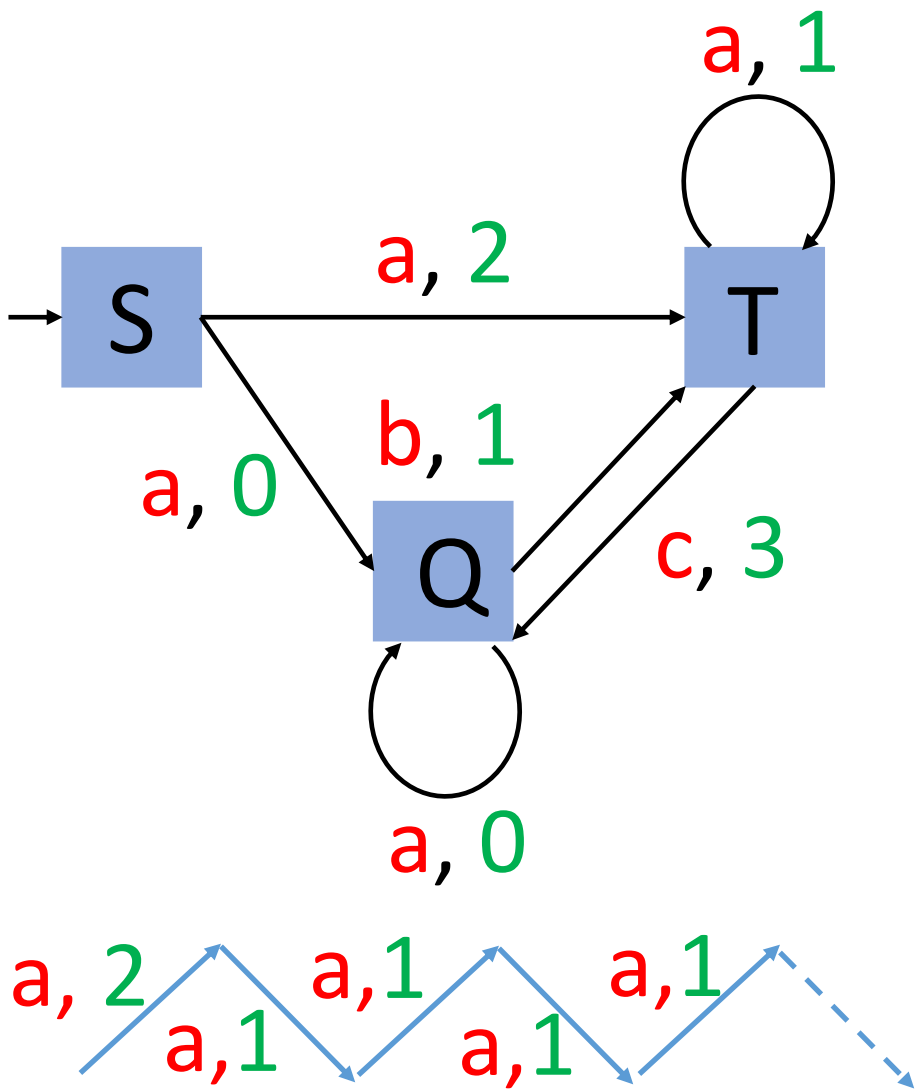
Aggregate function

- Aggregate function $f : \mathbb{N}^{\{\omega\}} \rightarrow \mathbb{R}$
- Commonly used aggregate functions
  - Limit supremum (Limsup)
  - Discounted-sum
  - Limit-average

Aggregate function

- Aggregate function $f : \mathbb{N}^{\{\omega\}} \to \mathbb{R}$
- Commonly used aggregate functions
  - Limit supremum (Limsup)
  - Discounted-sum
  - Limit-average

- Weight of run
  - Aggregate of its weight sequence
- Weight of word
  - Infimum/Supremum of weight of its runs



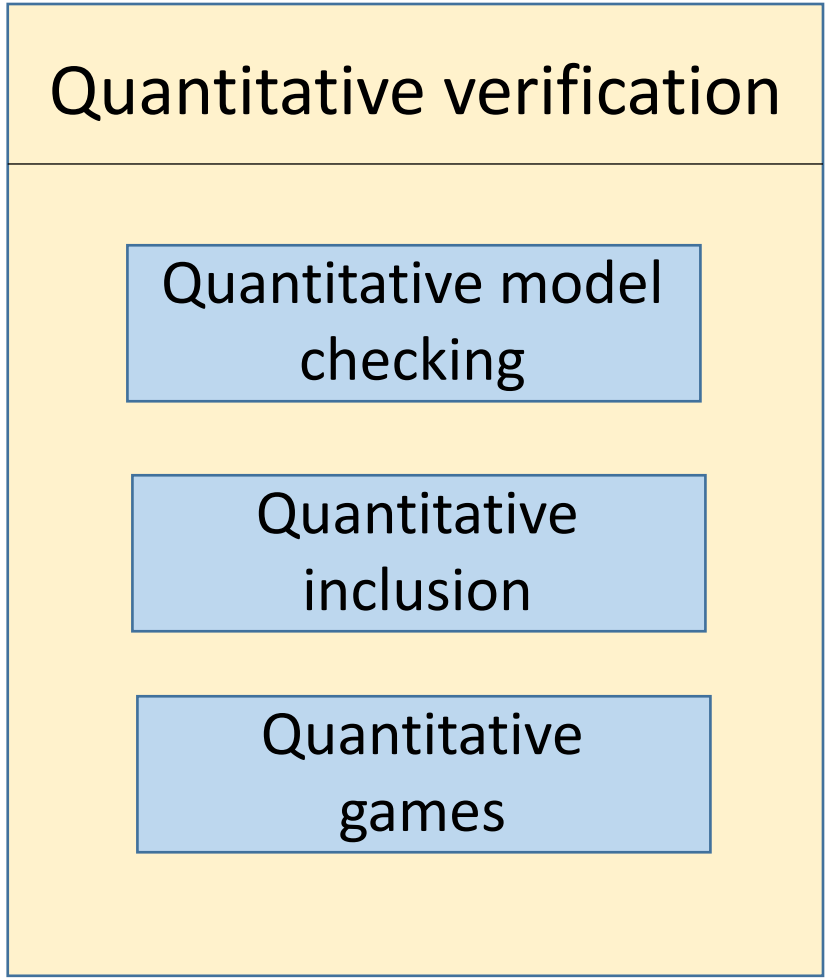
Comparison problem

- Quantitative verification deals with comparison of aggregates of weight sequences

- Given,
  - Infinite weight sequences A, B
  - Aggregate function $f : \mathbb{N}^{\{\omega\}} \rightarrow \mathbb{R}$
- **Comparison problem**
  - Is $f(A) \leq f(B)$?

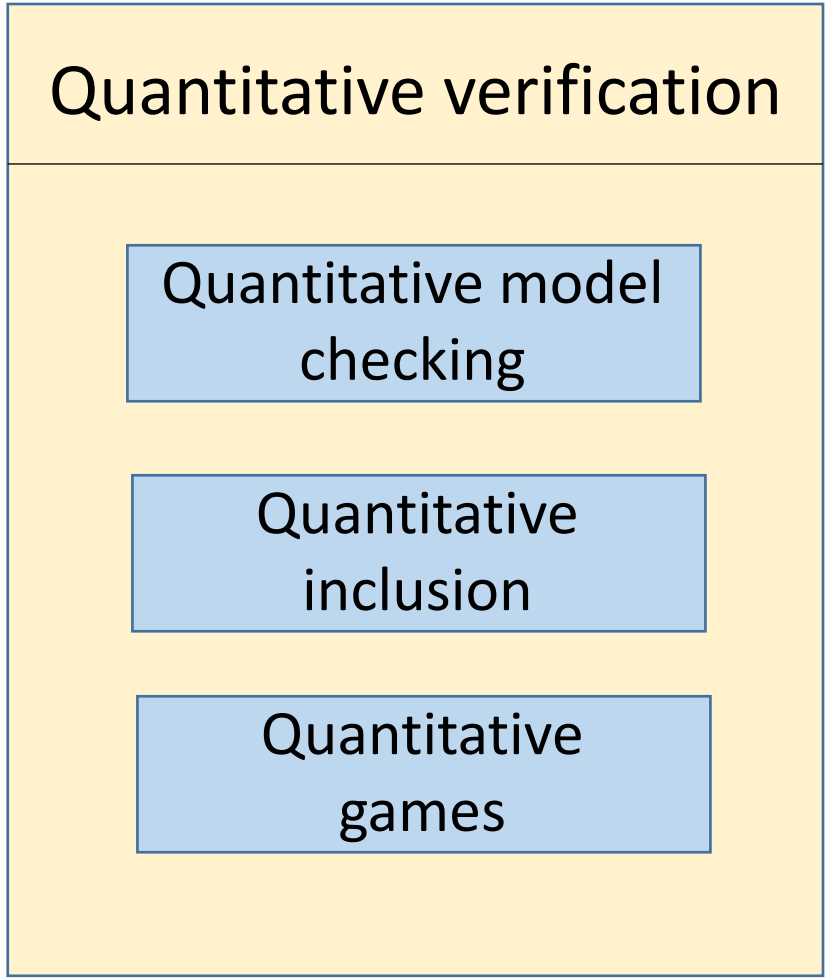
Comparison problem

- Drawbacks
  - Comparison dealt *ad hoc*
  - Different algorithm for same problem on different aggregate functions



Comparison problem

- Drawbacks
  - Comparison dealt *ad hoc*
  - Different algorithm for same problem on different aggregate functions
- Core contribution
  - Novel technique for comparison
  - Yields generic algorithms for a large class of aggregate functions
  - Obtains tighter complexity theoretic bounds



Comparator automata (Comparator)

- **Comparator for aggregate function** $f: \mathbb{N}^{\{\omega\}} \rightarrow \mathbb{R}$ is an automaton
  - Alphabet of comparator is $\{0,1,\ldots,\mu\}\times\{0,1,\ldots,\mu\}$
  - Accepts pair of bounded number sequences (A, B) iff $f(A) \leq f(B)$

Comparator automata (Comparator)

- **Comparator for aggregate function** $f: \mathbb{N}^{\{\omega\}} \rightarrow \mathbb{R}$ is an automaton
  - Alphabet of comparator is $\{0,1, \ldots, \mu\} \times \{0,1, \ldots, \mu\}$
  - Accepts pair of bounded number sequences (A, B) iff $f(A) \leq f(B)$

- **Büchi comparator**
  - When comparator for an aggregate function is a Büchi automata

Supremum function

- Supremum (Sup) of a number-sequence is the largest value in the sequence
  - $A = 1, 5, 3, 1^{\{\omega\}}$, $Sup(A) = 5$

Supremum function

- Supremum (Sup) of a number-sequence is the largest value in the sequence
  - $A = 1, 5, 3, 1^{\{\omega\}}$, $Sup(A) = 5$

- Supremum comparator is a Büchi comparator
  - States are (n, m)
    - Each records maximum value appearing on the sequence so far
  - Initial state $(0,0)$
  - Accepting state if $n \leq m$

Discounted-sum (DS) function

- Discounted-sum of number sequence R with discount factor $d > 1$ is

$$DS_d(R) = r_0 + \frac{r_1}{d} + \frac{r_2}{d^2} \ldots$$

- DS has applications in
  - Decaying rewards of agents in multi-agent systems [Abreu; ECONOMETRICA 1988]
  - Quantitative safety [Cerny, Henzinger, Radhakrishna; POPL 2013]

Discounted-sum (DS) function

- Discounted-sum of number sequence R with discount factor $d > 1$ is

$$DS_d(R) = r_0 + \frac{r_1}{d} + \frac{r_2}{d^2} \ldots$$

- DS has applications in
  - Decaying rewards of agents in multi-agent systems [Abreu; ECONOMETRICA 1988]
  - Quantitative safety [Cerny, Henzinger, Radhakrishna; POPL 2013]
- DS-comparator with discount-factor $d > 1$
  - Pair of weight sequences (A, B) is accepted by DS-comparator with discount-factor d iff $DS_d(A) \leq DS_d(B)$

Discounted-sum (DS) function

- Discounted-sum of number sequence R with discount factor $d > 1$ is

$$DS_d(R) = r_0 + \frac{r_1}{d} + \frac{r_2}{d^2} \ldots$$

- DS has applications in
  - Decaying rewards of agents in multi-agent systems [Abreu; ECONOMETRICA 1988]
  - Quantitative safety [Cerny, Henzinger, Radhakrishna; POPL 2013]
- DS-comparator with discount-factor $d > 1$
  - Pair of weight sequences (A, B) is accepted by DS-comparator with discount factor d iff $DS_d(A) \leq DS_d(B)$

Theorem: DS-comparator is Büchi when discount factor is an integer

DS-Comparator : Core Insight – I

- Sequence $A = (a_0, a_1, a_2 \ldots)$, integer discount factor $d > 1$
- $DS_d(A) = a_0 + \frac{a_1}{d} + \frac{a_2}{d^2} + \cdots$
  $= (a_0.a_1a_2 \ldots)_d = A_d$

→ Number in base d
[Chaudhuri, Sankaranarayanan, Vardi , LICS 2013]

DS-Comparator : Core Insight – II

- Sequence $A = (a_0, a_1, a_2 \ldots)$, integer discount factor $d > 1$
- $DS_d(A) = a_0 + \frac{a_1}{d} + \frac{a_2}{d^2} + \cdots$

  $= (a_0.a_1a_2 \ldots)_d = A_d$

- $DS_d(A) \leq DS_d(B)$ iff $A_d \leq B_d$
- So, there is a $C = (c_0, c_1, c_2, \ldots)$ such that
  - $DS_d(C) = C_d \geq 0$
  - $A_d + C_d = B_d$

→ Arithmetic in base d

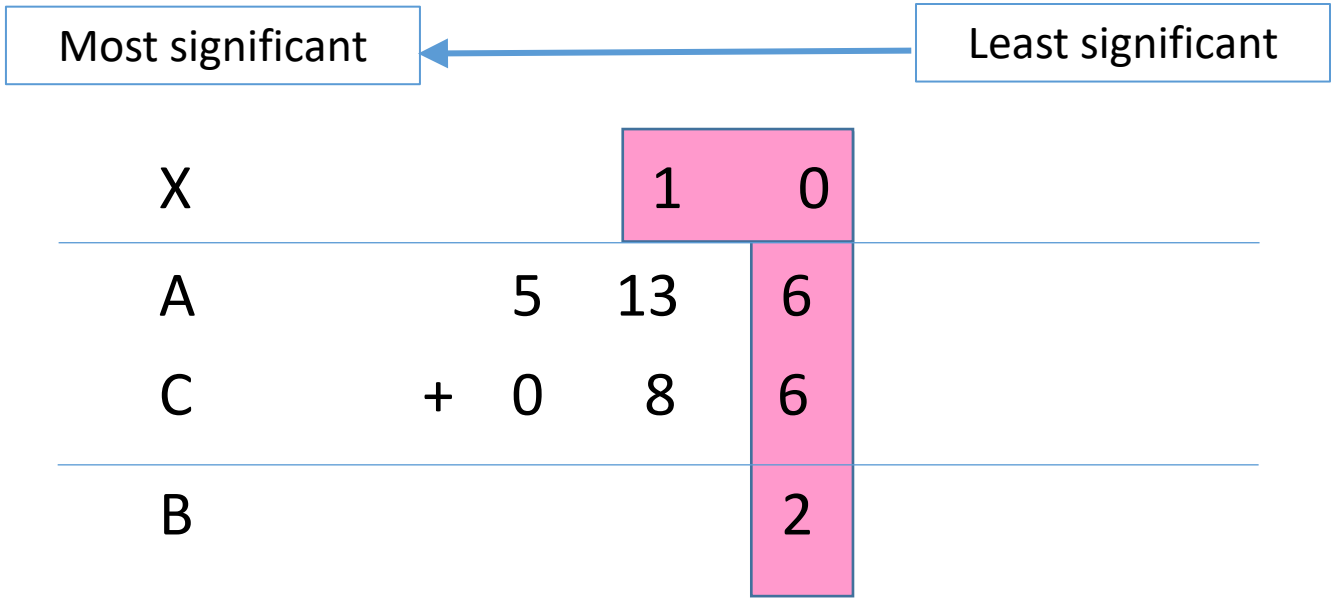
DS-Comparator : Core Insight – II (cont..)

- Consider ($d = 10$)

X				
A		5	13	6
C	+	0	8	6
B				

DS-Comparator : Core Insight – II (cont..)

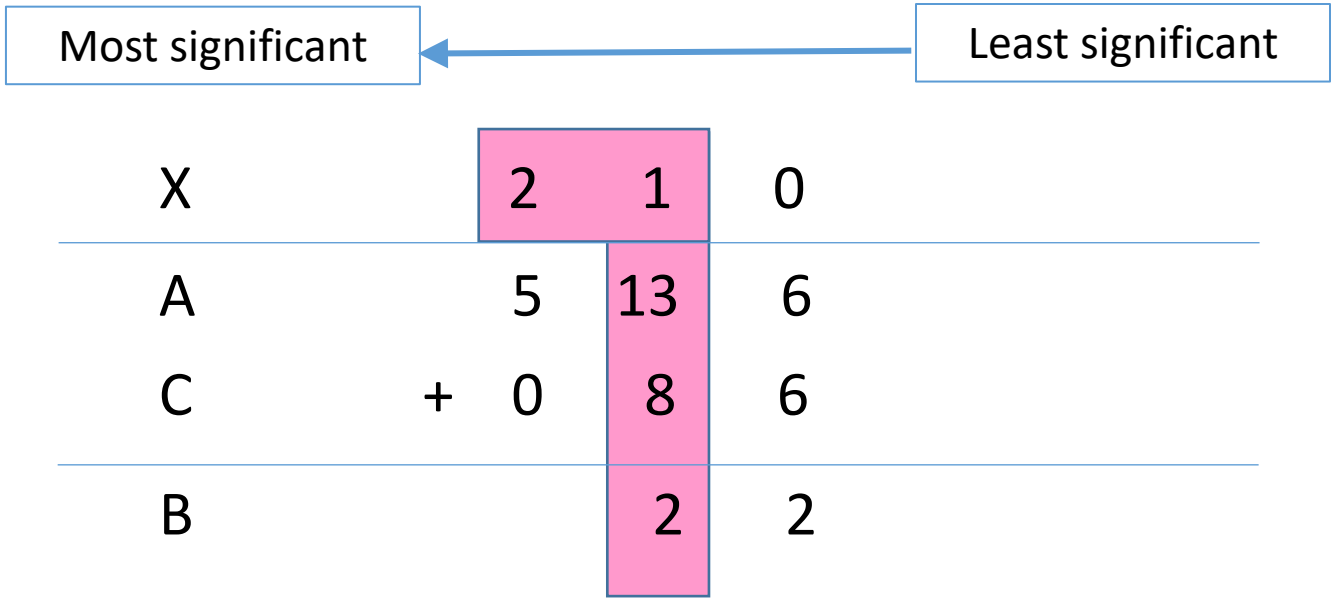
- Consider ($d = 10$)



$$i > 0, \quad a_i + c_i + x_i = b_i + d \cdot x_{\{i-1\}}$$

DS-Comparator : Core Insight – II (cont..)

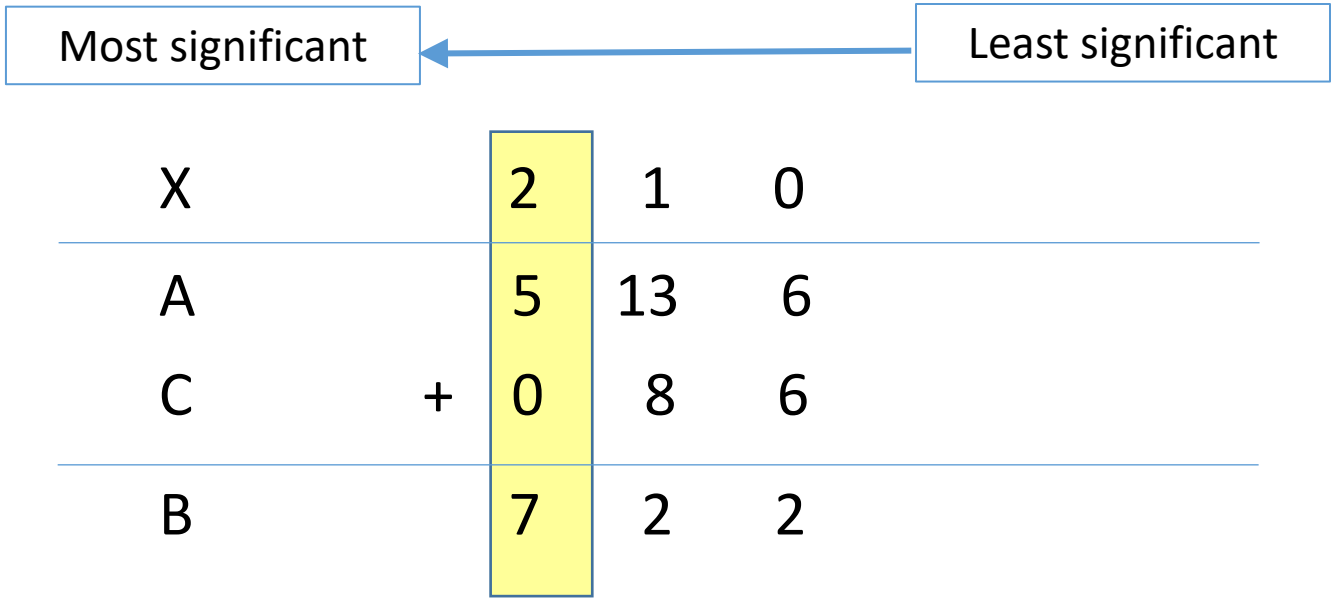
- Consider ($d = 10$)



$$i > 0, \quad a_i + c_i + x_i = b_i + d \cdot x_{\{i-1\}}$$

DS-Comparator : Core Insight – II (cont..)

- Consider ($d = 10$)



$i = 0, \quad a_0 + c_0 + x_0 = b_0$

$i > 0, \quad a_i + c_i + x_i = b_i + d \cdot x_{\{i-1\}}$

DS-Comparator : Core Insight – II (cont..)

- Consider ($d = 10$)

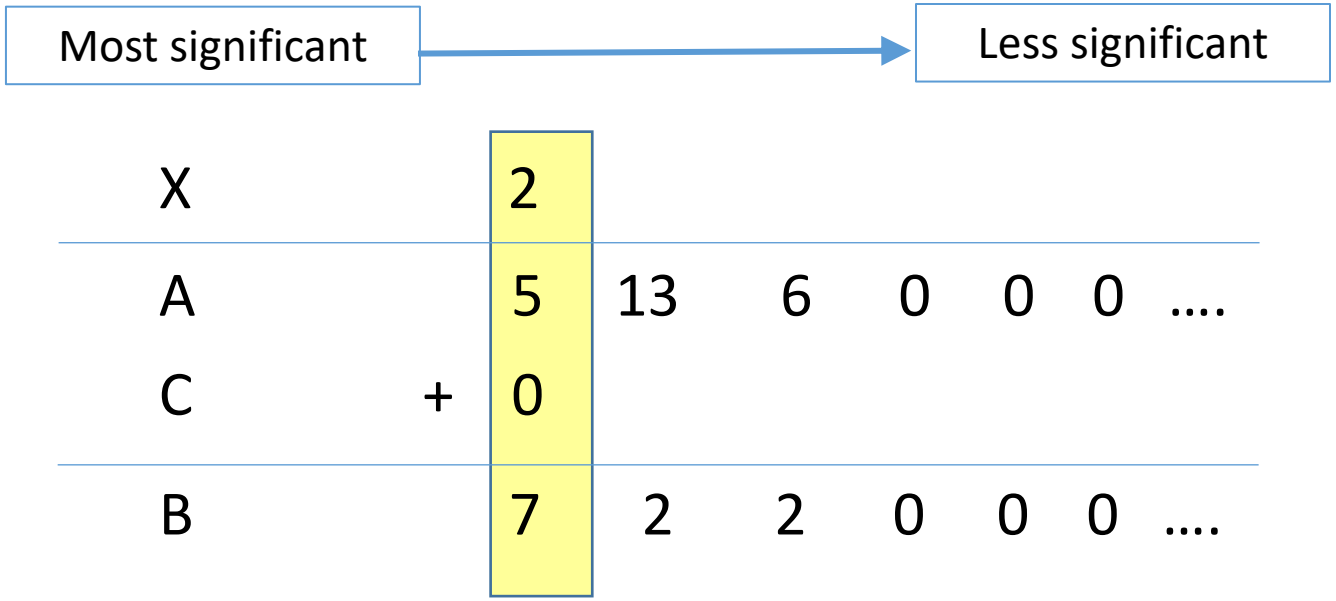
X							
A	5	13	6	0	0	0	….
C							
B	7	2	2	0	0	0	….

$i = 0, \quad a_0 + c_0 + x_0 = b_0$

$i > 0, \quad a_i + c_i + x_i = b_i + d \cdot x_{\{i-1\}}$

DS-Comparator : Core Insight – II (cont..)

- Consider ($d = 10$)

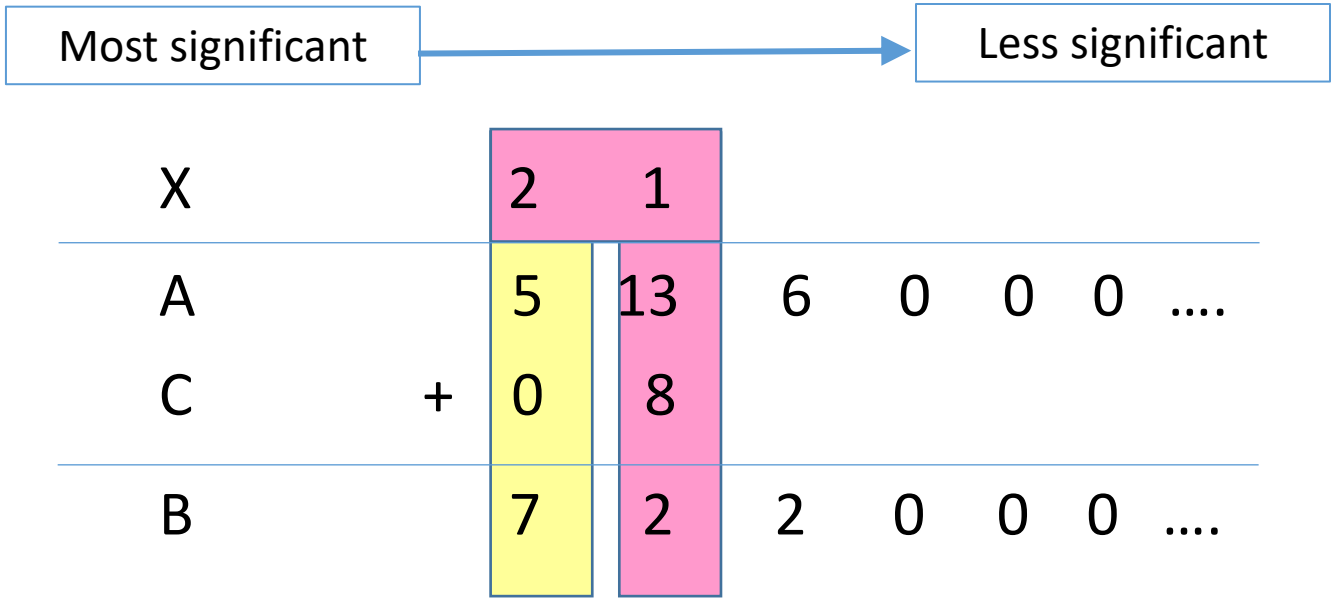


$i = 0, \quad a_0 + c_0 + x_0 = b_0$

$i > 0, \quad a_i + c_i + x_i = b_i + d \cdot x_{\{i-1\}}$

DS-Comparator : Core Insight – II (cont..)

- Consider ($d = 10$)

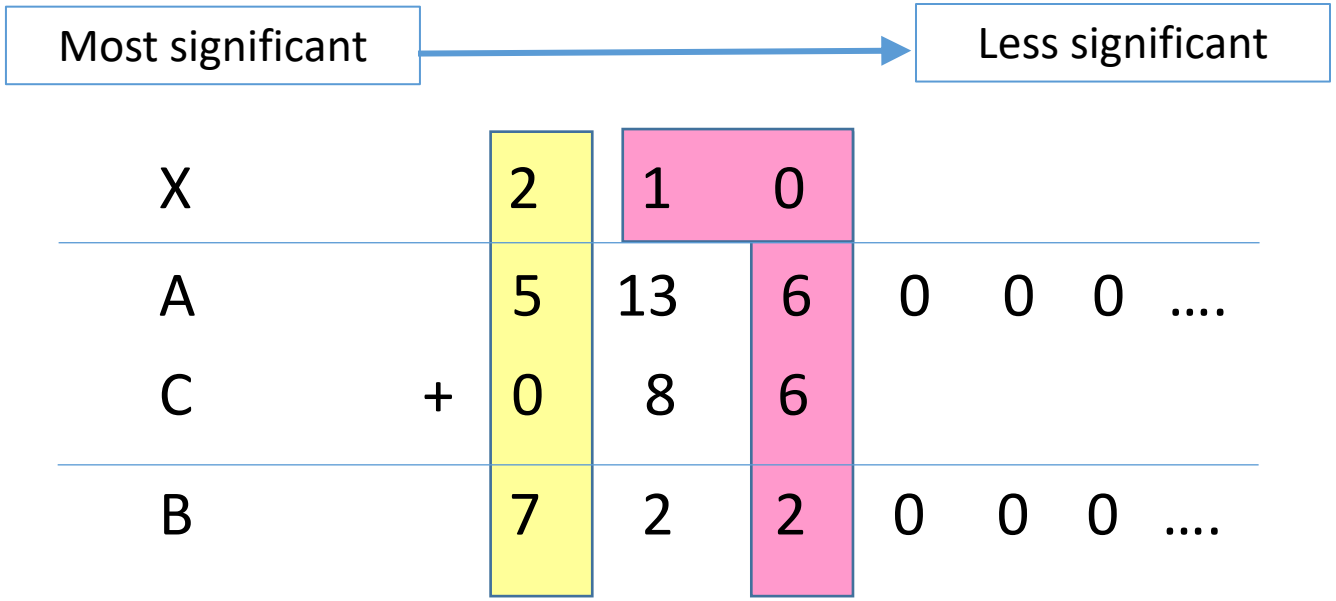


$i = 0, \quad a_0 + c_0 + x_0 = b_0$

$i > 0, \quad a_i + c_i + x_i = b_i + d \cdot x_{\{i-1\}}$

DS-Comparator : Core Insight – II (cont..)

- Consider ($d = 10$)

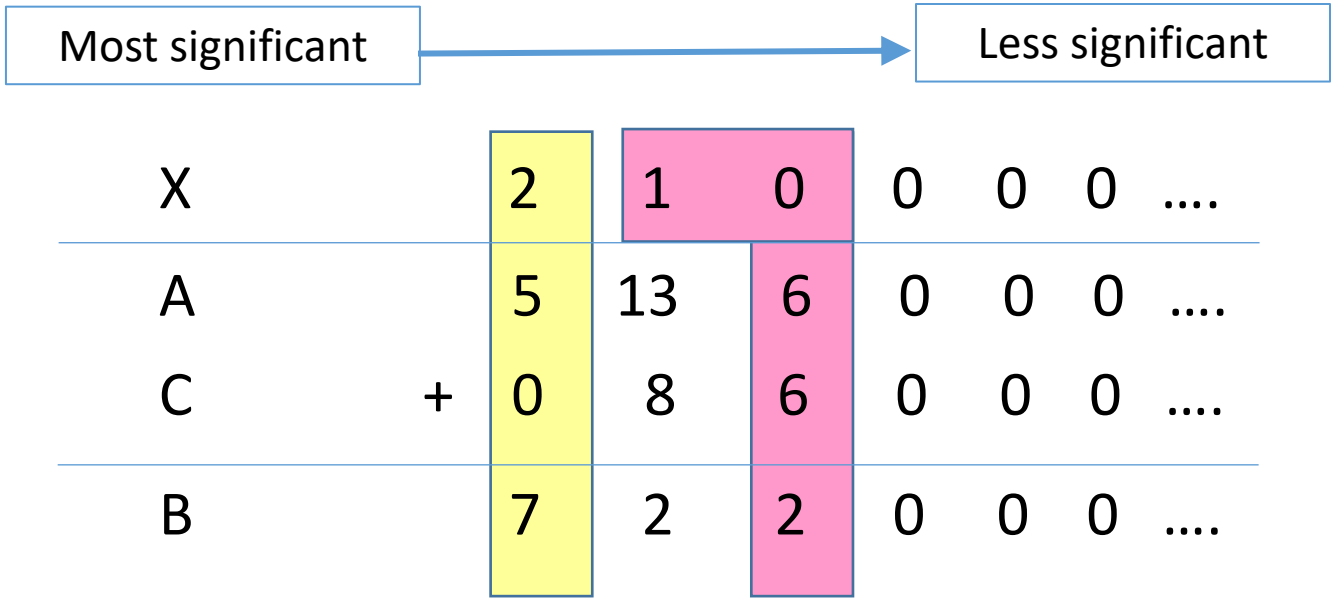


$i = 0, \quad a_0 + c_0 + x_0 = b_0$

$i > 0, \quad a_i + c_i + x_i = b_i + d \cdot x_{\{i-1\}}$

DS-Comparator : Core Insight – II (cont..)

- Consider ($d = 10$)

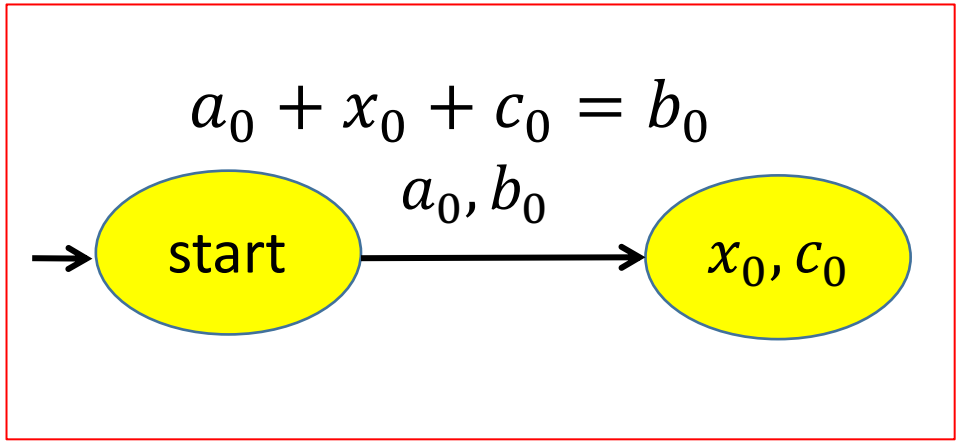


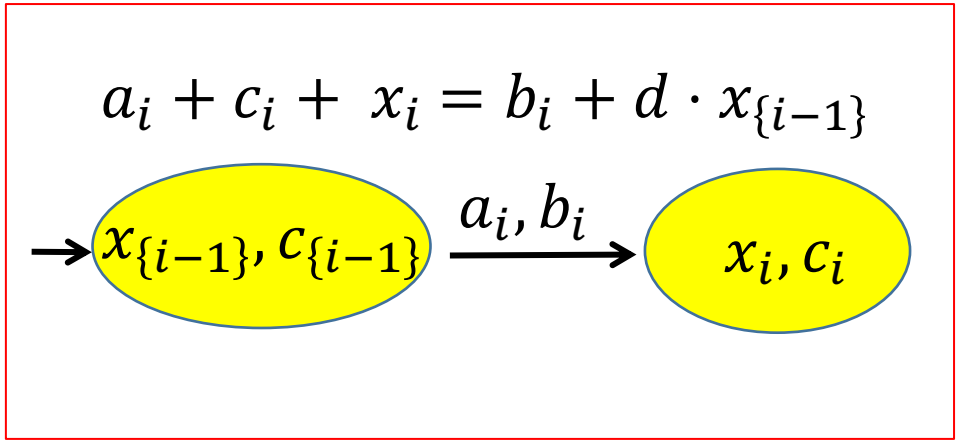
$i = 0, \quad a_0 + c_0 + x_0 = b_0$

$i > 0, \quad a_i + c_i + x_i = b_i + d \cdot x_{\{i-1\}}$

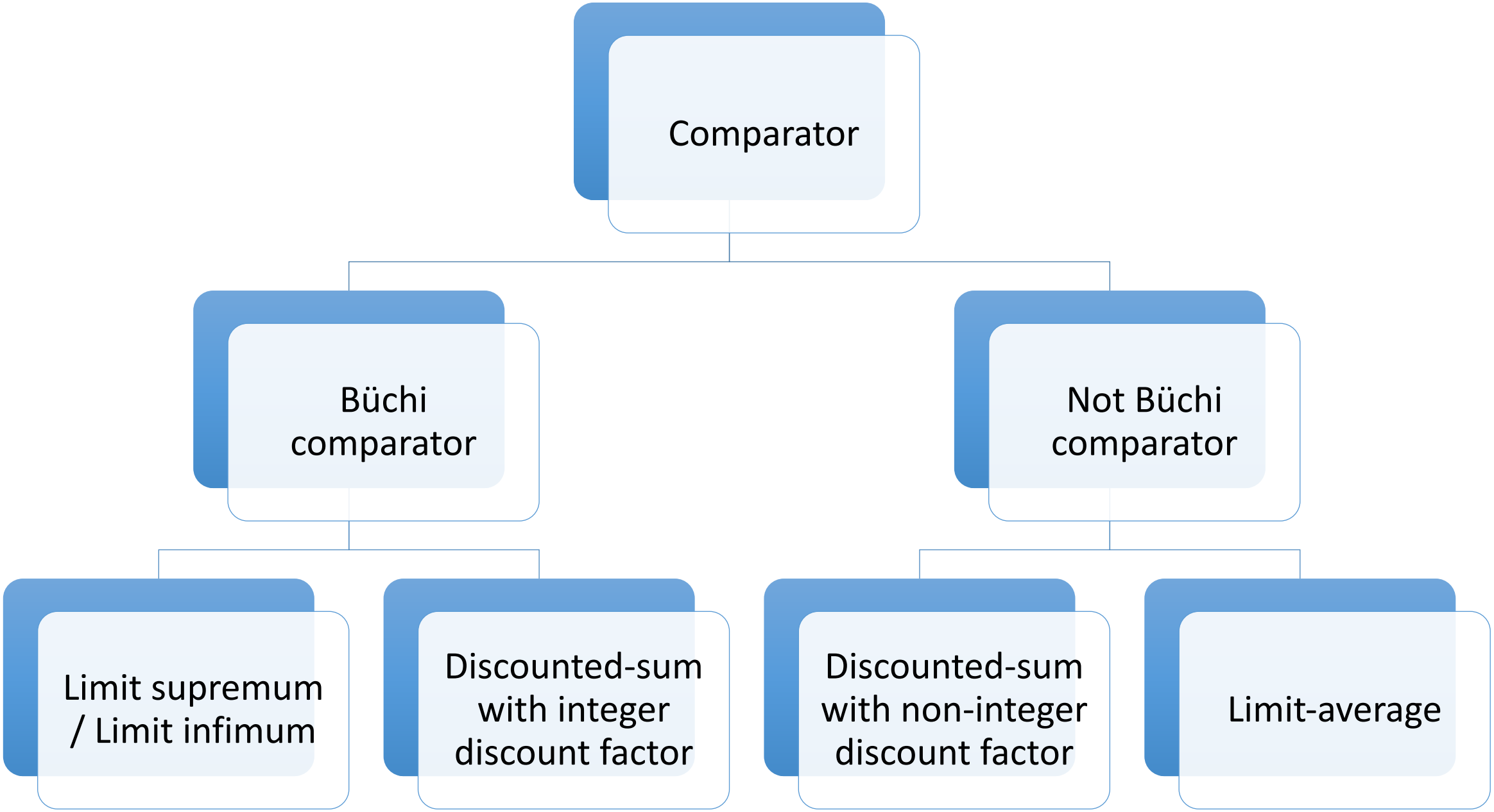
$\Rightarrow DS_d(A) + DS_d(C) = DS_d(B)$

DS-Comparator : Construction





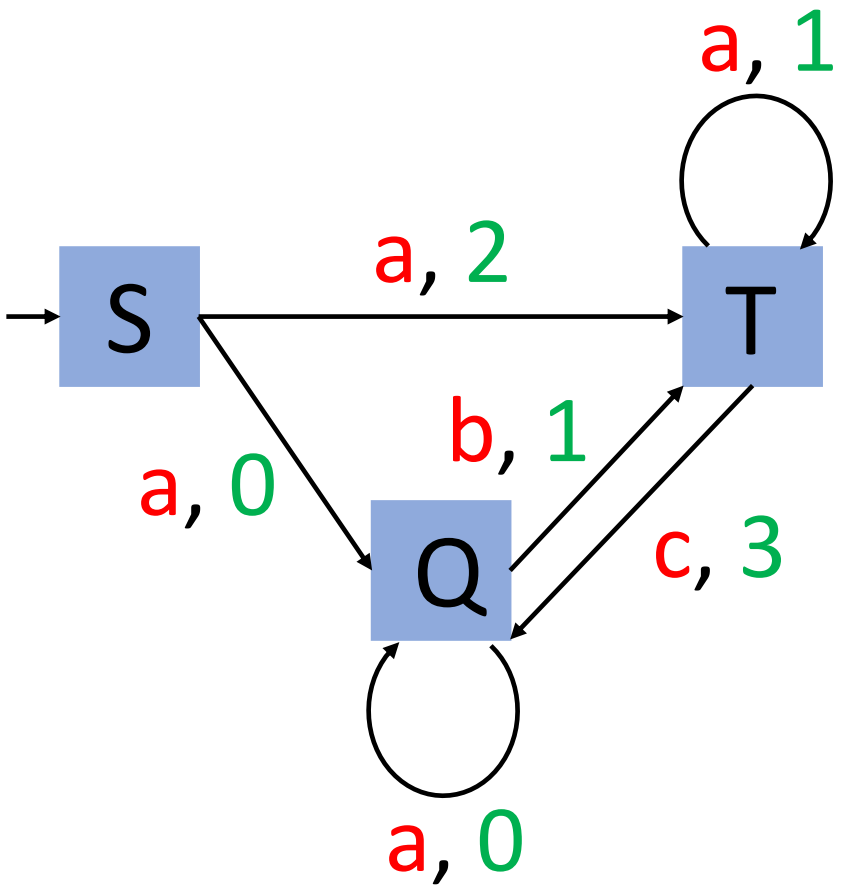
Automaton accepts (A, B) iff $DS_d(A) \leq DS_d(B)$


Comparator
Büchi comparator
Not Büchi comparator
Limit supremum / Limit infimum
Discounted-sum with integer discount factor
Discounted-sum with non-integer discount factor
Limit-average

Quantitative inclusion [Chatterjee, Doyen, Henzinger; CSL 2008]

- Quantitative finite-state machines
  - Weight of run
    - Aggregate of its weight sequence
    - Aggregate function $f : \mathbb{N}^{\{\omega\}} \to \mathbb{R}$
  - Weight of word
    - Infimum/Supremum of weight of its runs

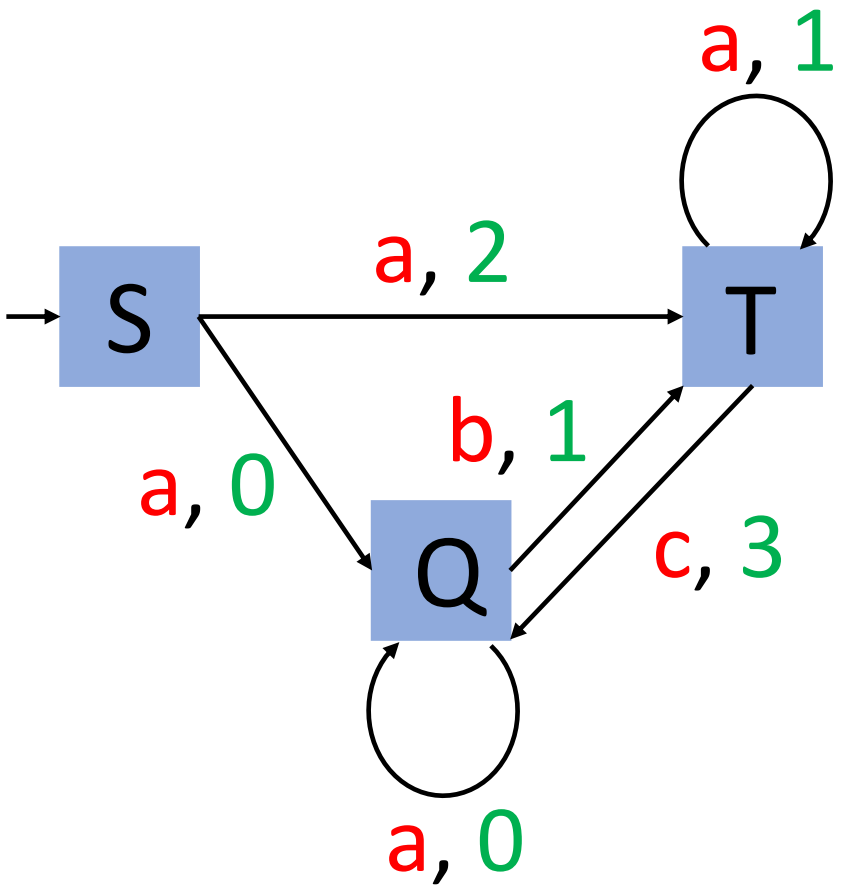
- Quantitative inclusion for aggregate function f between P and Q ($P \subseteq_f Q$)
  - Weight of every word in P is less than or equal to its weight in Q
  - Applied in quantitative model checking



Quantitative finite-state machine

Quantitative inclusion [Chatterjee, Doyen, Henzinger; CSL 2008]

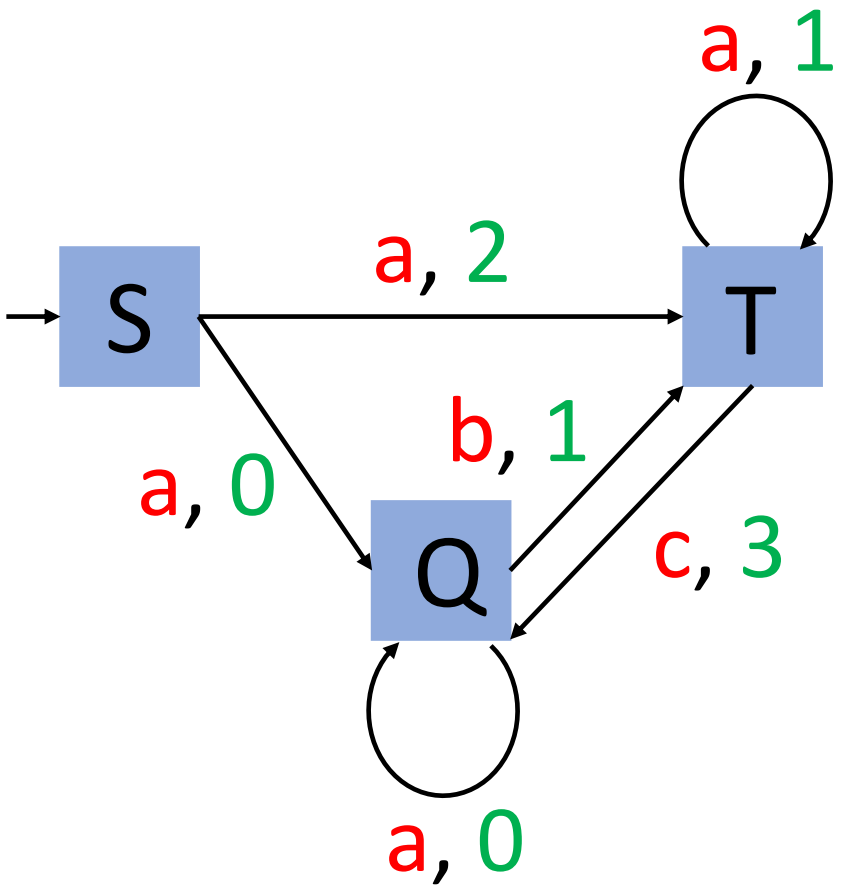
- Ad-hoc nature of quantitative inclusion
  - Supremum or limit-supremum [Chatterjee et al. CSL 2008]
  - Discounted-sum [Boker, Henzinger; LICS 2015] [Chatterjee et. al. CSL 2008]



Quantitative finite-state machine

Quantitative inclusion [Chatterjee, Doyen, Henzinger; CSL 2008]

- Ad-hoc nature of quantitative inclusion
  - Supremum or limit-supremum [Chatterjee et al. CSL 2008]
  - Discounted-sum [Boker, Henzinger; LICS 2015] [Chatterjee et. al. CSL 2008]
- Comparator-based generic algorithm
  - Applies when comparator is Büchi



Quantitative finite-state machine

Generic solution for $P \sqsubseteq_f Q$

- Reduced to language equivalence between Büchi automata
- Reduction is polynomial in size of P, Q and comparator C

Theorem: PSPACE in size of P, Q, C

DS-inclusion

- Quantitative inclusion with discounted sum aggregate function

- When discount factor is integer
  - EXPTIME upper bound [Boker, Henzinger; LICS 2015] [Chatterjee et. al. CSL 2008]
  - PSPACE lower bound via reduction from language inclusion

] Gap

DS-inclusion

- Quantitative inclusion with discounted sum aggregate function
- When discount factor is integer
  - EXPTIME upper bound [Boker, Henzinger; LICS 2015] [Chatterjee et. al. CSL 2008]
  - PSPACE lower bound via reduction from language inclusion

  ] Gap
- DS-comparator has $O(\mu^2/d)$ states
  - μ is maximum weight in P and Q
  - d is integer discount-factor

Theorem: DS-inclusion with integer discount factor is PSPACE-complete

Büchi comparator as a Parity objective

- 2-player game
  - Each player receives a reward in each round
  - Objective of player: To receive greater reward that the other player
- Winning objective can be defined using a Büchi comparator
  - Parity objective
- Use algorithms for Parity objective to solve game
  - Solve for games with incomplete-information with Parity objective [Kupferman and Vardi; Advances in Temporal Logic 2000]

Comparators, in a nutshell

- Comparator automata for the comparison problem
  - Generic algorithms for quantitative verification
    - Uses properties of Büchi automata
  - Comparator-based algorithms have tight complexity bounds
    - Better than ad-hoc algorithms in some cases

Comparators, in a nutshell

- Comparator automata for the comparison problem
  - Generic algorithms for quantitative verification
    - Uses properties of Büchi automata
  - Comparator-based algorithms have tight complexity bounds
    - Better than ad-hoc algorithms in some cases
- Symbolic quantitative reasoning?
  - *In practice* performance of comparator-based algorithms vs. traditional algorithms
- Applications of non-Büchi comparators
  - Büchi pushdown comparators