# A Weak Bisimulation for Weighted Automata

Peter Kemper
College of William and Mary

- Weighted Automata and Semirings
  - here focus on commutative & idempotent semirings
- Weak Bisimulation
- Composition operators
- Congruence property

# Motivation

Notions of equivalence have been detected for many notations:

- process algebras
- automata
- stochastic processes

Equivalences are useful

- for a theoretical investigation of equivalent behaviour
- increasing the efficiency of analysis techiques by
  - minimization to the smallest equivalent automaton
  - composition of minimized automata

  requires congruence property!

Many different equivalences exist:

trace-equivalence, failure equivalence, strong / weak bisimulation, ...

We consider a weak bisimulation for automata whose nodes and edges are annotated by labels and weights.

Weights are elements of an algebra -> a semiring.

# Semiring

- Semiring $K_{+,*} = (K,+,*,0,1)$
  Operations + and * defined for K have the following properties
  - associative: + and *
  - commutative: +
  - right/left distributive for + with respect to *
  - 0 and 1 are additive and multiplicative identities with $0 \neq 1$
  - for all $k \in K \quad 0 * k = k * 0 = 0$

- What is so special?
  Similar to a ring, but each element need not(!) have an additive inverse.
- Special cases:
  - Idempotent semiring (or Dioid): + is idempotent: a+a=a
  - Commutative semiring: * is commutative

# Semiring

Alternative definition

A semiring is a set $K$ equipped with two binary operations + and ·, called addition and multiplication, such that:

- (K, +) is a commutative monoid with identity element 0:
  - $(a + b) + c = a + (b + c)$
  - $0 + a = a + 0 = a$
  - $a + b = b + a$
- (K, ·) is a monoid with identity element 1:
  - $(a \cdot b) \cdot c = a \cdot (b \cdot c)$
  - $1 \cdot a = a \cdot 1 = a$
- Multiplication distributes over addition:
  - $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$
  - $(a + b) \cdot c = (a \cdot c) + (b \cdot c)$
- 0 annihilates K:
  - $0 \cdot a = a \cdot 0 = 0$

# Semiring

- Semiring $K_{+,*} = (K, +, *, 0, 1)$

- Examples
  - Boolean semiring $(B, \vee, \wedge, 0, 1)$
  - Real numbers $(R, +, *, 0, 1)$
  - max/+ semiring $(R \cup -\infty, \max, +, -\infty, 0)$
  - min/+ semiring $(R \cup \infty, \min, +, \infty, 0)$
  - max/min semiring $(R \cup -\infty \cup \infty, \max, \min, -\infty, \infty)$
  - square matrices $(R^{n \times n}, +, *, 0, 1)$

  - A Kleene algebra is an idempotent semiring R with an additional unary operator * : R → R called the Kleene star. Kleene algebras are important in the theory of formal languages and regular expressions.

# Idempotent Semiring

- Let's define a partial order $\leq$ on an idempotent semiring:

  $a \leq b$ whenever $a + b = b$

  (or, equivalently, if there exists an x such that $a + x = b$).
- Observations:
  - 0 is the least element with respect to this order:

    $0 \leq a$ for all a.
  - Addition and multiplication respect the ordering :

    $a \leq b$ implies

    $ac \leq bc$

    $ca \leq cb$

    $(a+c) \leq (b+c)$

# Kleene Algebra

A Kleene algebra is a set A with two binary operations $+ : A \times A \to A$ and $\cdot : A \times A \to A$ and one function $* : A \to A$, (Notation: a+b, ab and a*) and

- Associativity of + and ·, Commutativity of +
- Distributivity of · over +
- Identity elements for + and ·:

  exists 0 in A such that for all a in A: $a + 0 = 0 + a = a$.

  exists 1 in A such that for all a in A: $a1 = 1a = a$.
- $a0 = 0a = 0$ for all a in A.

The above axioms define a semiring.

We further require:

- + is idempotent: $a + a = a$ for all a in A.

# Kleene Algebra

- Let's define a partial order ≤ on A:
  a ≤ b if and only if a + b = b
  (or equivalently: a ≤ b if and only if exists x in A such that a + x = b).

With this order we can formulate the last two axioms about the operation *:

- 1 + a(a*) ≤ a* for all a in A.
- 1 + (a*)a ≤ a* for all a in A.
- if a and x are in A such that ax ≤ x, then a*x ≤ x
- if a and x are in A such that xa ≤ x, then x(a*) ≤ x

Think of
  a + b as the "union" or the "least upper bound" of a and b and of
  ab as some multiplication which is monotonic, in the sense that a ≤ b implies ax ≤ bx.

The idea behind the star operator is a* = 1 + a + aa + aaa + ...

From the standpoint of programming theory, one may also interpret + as "choice", · as "sequencing" and * as "iteration".

- Example: Set of regular expressions over a finite alphabet

# Weighted Automaton

A finite K-Automaton over finite alphabet L (including $\tau$ ) is $A = (S, \alpha, T, \beta)$
with S : finite set of states and maps
giving initial, transition and final weights.

$$\alpha : S \to K,$$
$$T : S \times L \times S \to K,$$
$$\beta : S \to K$$

E.g. weights interpreted as costs, distances, time, ...
Weights multiply along a path, sum up over different paths.

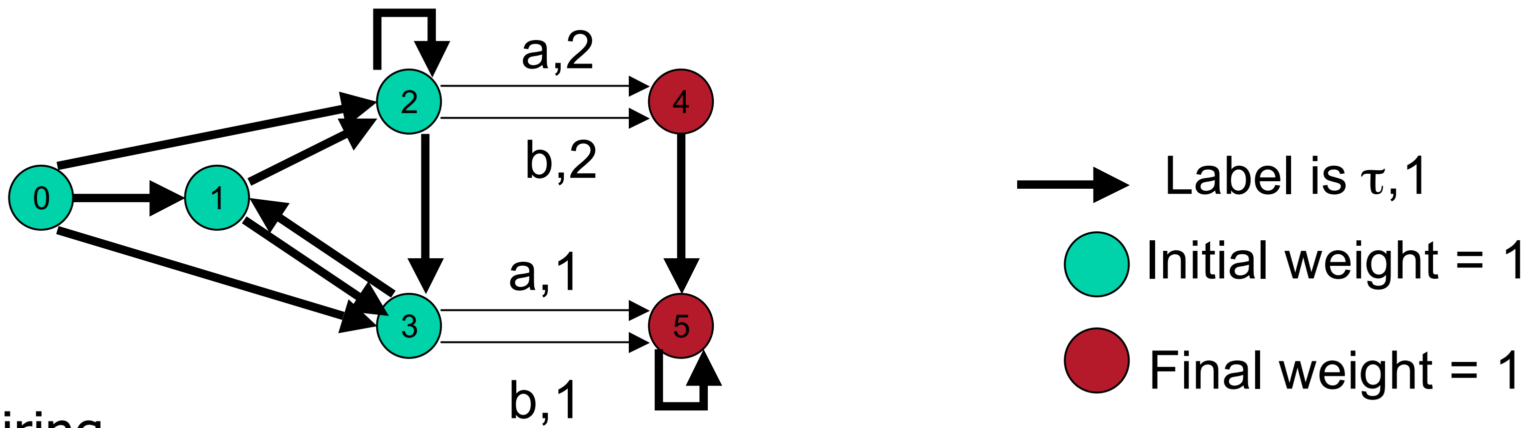
We focus on commutative and idempotent K-automata, i.e.,
K is a semiring where * is commutative and + is idempotent!
Examples

- Boolean semiring
- max/+ semiring
- min/+ semiring
- max/min semiring

Transitions are described by matrices
Idempotency implies:

$$\sum_{k=0}^{\infty} \mathbf{A}^{\mathbf{k}} = \sum_{k=0}^{\infty} \mathbf{A}^{\mathbf{k}} \cdot \sum_{k=0}^{\infty} \mathbf{A}^{\mathbf{k}}$$

# Examples




- Boolean semiring,
  - weights encode existence / non-existence of paths in directed graphs
  - labels serve the same purpose, hence weights are usually omitted
  - idempotency is quite natural:
    - existence of a paths remains valid in case of multiple paths
- Max/+ semiring
  - interpretation
    - weights are multiplied along a path, * is +, weight of a path is the sum over all edge weights
    - sum over all paths starting at a node is given by max, hence the path with highest weight is taken (snob if these are costs, greedy if this is profit)
- Max/Min semiring
  - interpretation
    - weight of a path: * is min, weight of a path gives minimal weight of its edges
    - sum over paths: + is max, selects path whose bottleneck has largest capacity

# Some more notation

- Weight of path $\pi$ $\quad w(\pi) = \alpha(s_0) \cdot \left( \prod_{i=1}^{n} T(s_{i-1}, l_i, s_i) \right) \beta(s_n)$

  or by vectors/matrices $\quad = \mathbf{a}(s_0) \left( \prod_{i=1}^{n} \mathbf{M_{li}}(s_{i-1}, s_i) \right) \mathbf{b}(s_n)$

- Weight of sequence $\sigma$ $\quad w(\sigma) = \mathbf{a} \cdot \left( \prod_{i=1}^{n} \mathbf{M}_{li} \right) \mathbf{b}$

- Define automaton A* where sequences of $\tau$-transitions are replaced by single $\varepsilon$ transition.

$$\mathbf{M}_{\varepsilon}' = \mathbf{M}_{\tau}^{*} = \sum_{i=0}^{\infty} \mathbf{M}_{\tau}^{i}, \qquad \mathbf{M}_{l}' = \mathbf{M}_{\varepsilon}' \cdot \mathbf{M}_{l} \cdot \mathbf{M}_{\varepsilon}', \quad \mathbf{b}' = \mathbf{M}_{\varepsilon}' \cdot \mathbf{b}$$

- Weight of sequence $\sigma'$

$$w'(\sigma') = \mathbf{a} \cdot \left( \prod_{i=1}^{n} \mathbf{M}_{li}' \right) \mathbf{b}'$$

$$= \mathbf{a} \cdot \left( \prod_{i=1}^{n} \mathbf{M}_{\varepsilon}' \mathbf{M}_{li} \cdot \mathbf{M}_{\varepsilon}' \right) \mathbf{M}_{\varepsilon}' \cdot \mathbf{b}$$

$$= \mathbf{a} \cdot \left( \prod_{i=1}^{n} \mathbf{M}_{\varepsilon}' \mathbf{M}_{li} \right) \mathbf{M}_{\varepsilon}' \cdot \mathbf{b}$$

# Weak bisimulation of K-automata

An equivalence relation $R \subseteq S \times S$ is a weak bisimulation relation if $\text{for all } (s_1, s_2) \in R, \text{all } l \in L \setminus \{\tau\} \cup \{\varepsilon\}, \text{all equivalence classes } C \in S / R$

$$\alpha(s_1) = \alpha(s_2)$$
$$\beta'(s_1) = \beta'(s_2)$$
$$T'(s_1, l, C) = T'(s_2, l, C)$$

or in terms of matrices

$$\mathbf{a}(s_1) = \mathbf{a}(s_2)$$
$$\mathbf{b}'(s_1) = \mathbf{b}'(s_2)$$
$$\mathbf{M}'_l(s_1, C) = \mathbf{M}'_l(s_2, C)$$

Two states are weakly bisimilar, $s_1 \approx s_2$ , if $(s_1, s_2) \in R$

Two automata are weakly bisimilar, $A_1 \approx A_2$ , if there is a weak bisimulation on the union of both automata such that

$$\alpha(\mathrm{C}_1) = \alpha(\mathrm{C}_2) \text{ for all } C \in S / R$$

# Theorem

$$\text{If } A_1 \approx A_2 \text{ for Ki-Automata } A_1, A_2 \text{ then } \mathrm{w}_1{}'(\sigma) = \mathrm{w}_2{}'(\sigma)$$
$$\text{for all } \sigma \in L'^* \text{ where } L' = (L_1 \cup L_2) \setminus \{\tau\} \cup \{\varepsilon\}$$

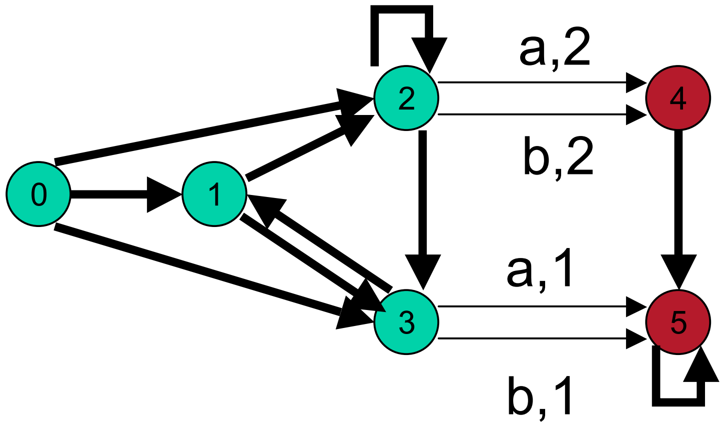
Weights of sequences are equal in weakly bisimilar automata.

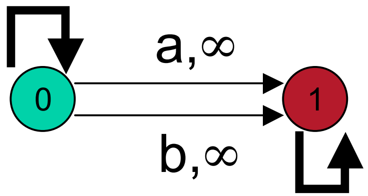
Ki ? commutative and idempotent semiring K

Sequence? sequence considers all paths that have same sequence of labels, may start or stop at any state

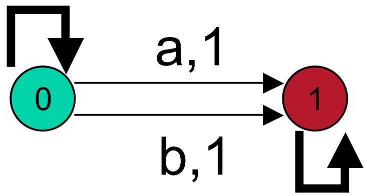
Weakly ? Paths can contain subpaths of $\tau$-labeled transitions represented by a single $\varepsilon$-labeled transition.

# Example

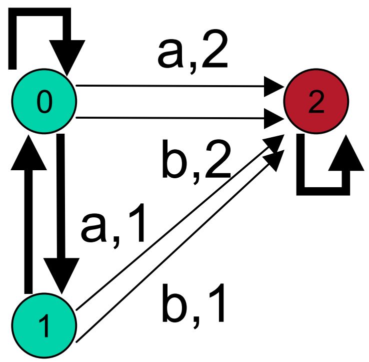



is weakly bisimilar for max/+ semiring




is weakly bisimilar for min/max semiring




is weakly bisimilar for min/+ and max/min semiring

# Deloping further

- consider largest bisimulation, i.e. the one with fewest classes
  - same argumentation as for Milner´s CCS
- computation by O(nm) fix point algorithm, n states, m edges
  - starting from boolean semiring as in the concurrency workbench (Cleaveland, Parrows, Steffen)
  - extended to semiring of real numbers by Buchholz
  - extension to more general semirings straightforward
  - more efficient ones like O(n log m) as for boolean semiring ???
  - presupposes also computation of A*
- bisimulation useful if preserved by composition operations (congruence property)
  - composition operations for automata ?

    sum

    direct or cascaded product

    synchronized product

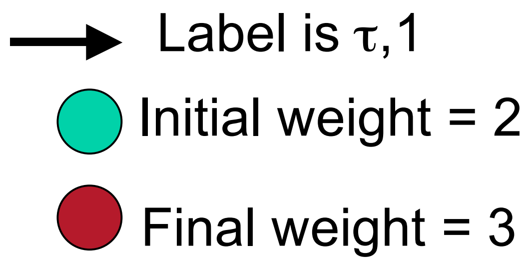
    specific type of choice

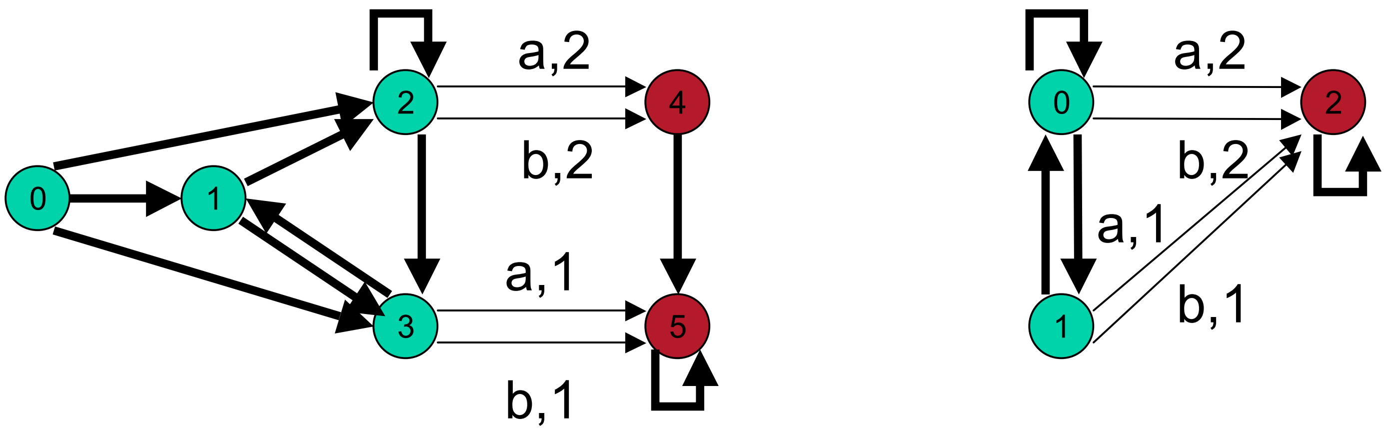
good news: these are all ok !!!

but how are they defined ?

# Composition operations

- Sum
  - union of automata with no interaction
- Direct or cascaded product
  - build union of state sets and labels
  - take initial weights only from first automaton
  - take final weights only from second automaton
  - connect first with second automaton by new $\tau$-transitions between final states of first, initial states of second automaton



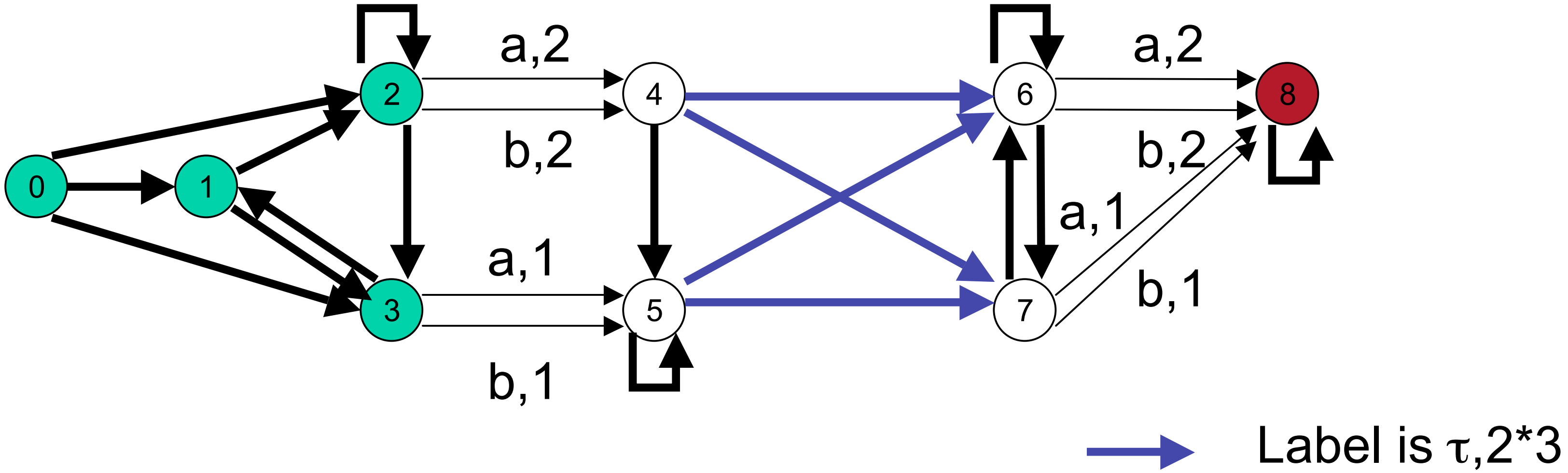
# Composition operations

- Sum
  - union of automata with no interaction
- Direct or cascaded product
  - build union of state sets and labels
  - take initial weights only from first automaton
  - take final weights only from second automaton
  - connect first with second automaton by new tau-transitions between final states of first, initial states of second automaton

Label is τ,1

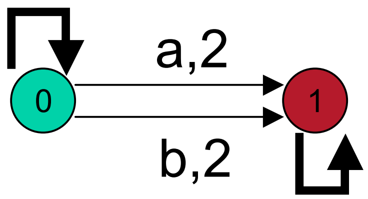
Initial weight = 2

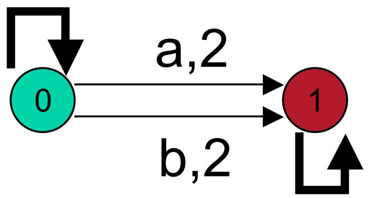
Final weight = 3



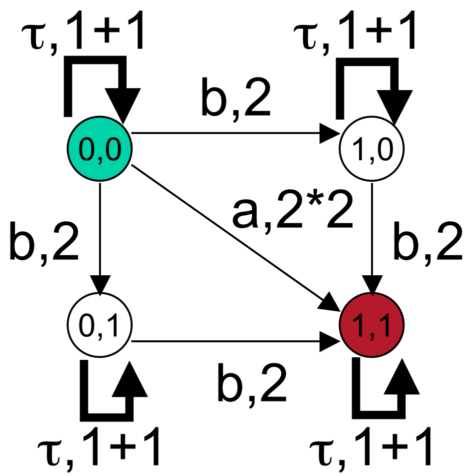
# Composition operations

- Synchronized product (with subset of labels for synchronisation)
  - build cross product of state sets, union of label sets
  - take product of initial weights
  - take product of final weights
  - take product of transition weights in case of synch otherwise proceed independently
  - Note: free product is special case with empty set of labels for synchronisation



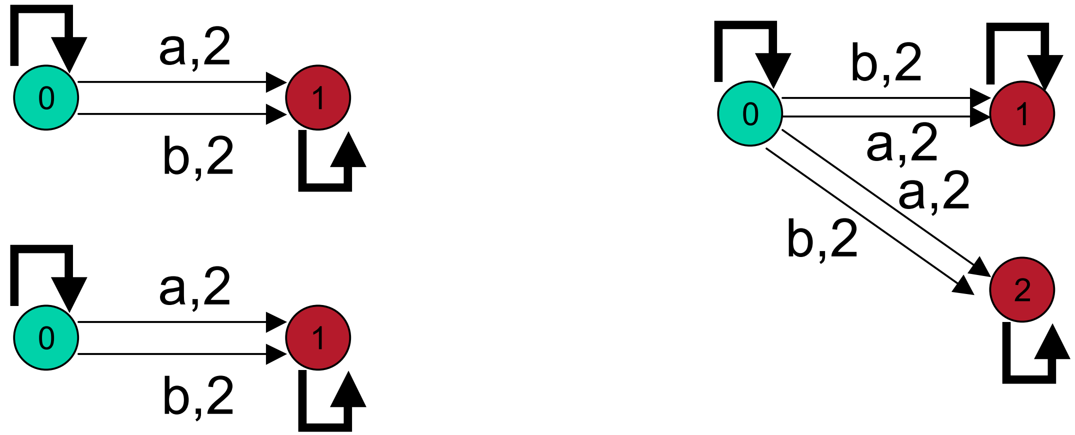




Synch on {a}



# Composition operations

- Synchronized product (with subset of labels for synchronisation)
  - build cross product of state sets, union of label sets
  - take product of initial weights
  - take product of final weights
  - take product of transition weights in case of synch otherwise proceed independently
  - Note: free product is special case with empty set of labels for synchronisation
- Choice
  - connects automata by merging only initial states
  - initial states must be unique and have initial weight 1 and equal final weights



# Theorem

If $A_1 \approx A_2$ and $A_3$ are finite Ki - Automata then

1. $A_1 + A_3 \approx A_2 + A_3$ — direct sum
2. $A_1 \cdot A_3 \approx A_2 \cdot A_3$ and $A_2 \cdot A_3 \approx A_1 \cdot A_3$ — direct product
3. $A_1 \|_{L_C} A_3 \approx A_2 \|_{L_C} A_3$ and $A_3 \|_{L_C} A_1 \approx A_3 \|_{L_C} A_2$ — synchronized product

and if choice is defined then

4. $A_1 \vee A_3 \approx A_2 \vee A_3$ and $A_3 \vee A_1 \approx A_3 \vee A_2$ — choice

Some notes on proofs:

- proofs are lengthy,
- argumentation based matrices helps,
- argumentation along paths, resp. sequences more tedious
- idempotency simplifies valuation for concatenation of $\tau^* l \, \tau^*$ transitions
- note that algebra does not provide inverse elements wrt + and *

# Summary

- Weak Bisimulation for weighted automata over commutative and idempotent semirings
- Congruence for
  - sum
  - direct or cascaded product
  - synchronized product
  - specific choice operator

References:

P.Buchholz,P.Kemper: Weak bisimulation for (max/+) automata and related models; Journal of Automata, Languages and Combinatorics, Vol 8, Number 2, 2003.

P.Buchholz,P.Kemper: Quantifying the dynamic behaviour of process algebras; Proc. joint PAPM/ProbMIV w´shop, Springer LNCS 2165, 2001.