A Type System for Weighted Automata and Rational Expressions

Akim Demaille[1], Alexandre Duret-Lutz[1], Sylvain Lombardy[2], Luca Saiu[1,3], Jacques Sakarovitch[3]

[1]EPITA/LRDE, [2]LaBRI, Université de Bordeaux, [3]LTCI, CNRS / Télécom-ParisTech

http://vaucanson.lrde.epita.fr
http://vaucanson-project.org

CIAA 2014; August, 1st 2014

(2014-08-03 00:28:28 +0200 a998c33)

Vaucanson

General-purpose platform for weighted automata and transducers.

- Genericity
  - Acceptors and transducers
  - Rational expressions
  - Weights: Boolean, Usual, Tropical...
  - Labels: letter, word, ε...
  - Letters: `chars`, `ints`...
- Performance
  - C++ templated library
  - No dynamic polymorphism (`virtual`)
  - Efficient algorithms and data structures
- Flexibility
  - A dynamic API on top of the templated library
  - A dynamic type system for automata, rational expressions, etc.
  - An interactive GUI on top of IPython

Vaucanson

General-purpose platform for weighted automata and transducers.

- Genericity
  - Acceptors and transducers
  - Rational expressions
  - Weights: Boolean, Usual, Tropical...
  - Labels: letter, word, ε...
  - Letters: `char`s, `int`s...
- Performance
  - C++ templated library
  - No dynamic polymorphism (`virtual`)
  - Efficient algorithms and data structures
- Flexibility
  - A dynamic API on top of the templated library
  - A dynamic type system for automata, rational expressions, etc.
  - An interactive GUI on top of IPython

Vaucanson

General-purpose platform for weighted automata and transducers.

- Genericity
  - Acceptors and transducers
  - Rational expressions
  - Weights: Boolean, Usual, Tropical...
  - Labels: letter, word, ε...
  - Letters: `char`s, `int`s...
- Performance
  - C++ templated library
  - No dynamic polymorphism (`virtual`)
  - Efficient algorithms and data structures
- Flexibility
  - A dynamic API on top of the templated library
  - A dynamic type system for automata, rational expressions, etc.
  - An interactive GUI on top of IPython

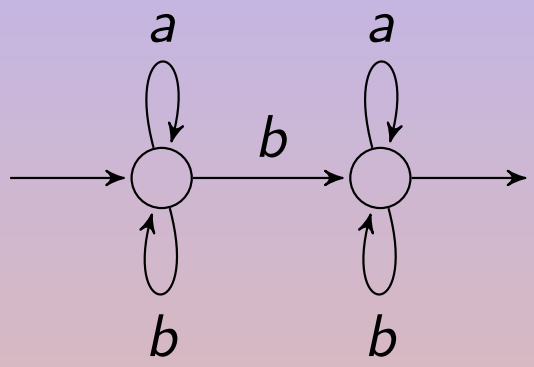
Vaucanson

General-purpose platform for weighted automata and transducers.

- Genericity
  - Acceptors and transducers
  - Rational expressions
  - Weights: Boolean, Usual, Tropical...
  - Labels: letter, word, ε...
  - Letters: `char`s, `int`s...
- Performance
  - C++ templated library
  - No dynamic polymorphism (`virtual`)
  - Efficient algorithms and data structures
- Flexibility
  - A dynamic API on top of the templated library
  - A dynamic type system for automata, rational expressions, etc.
  - An interactive GUI on top of IPython

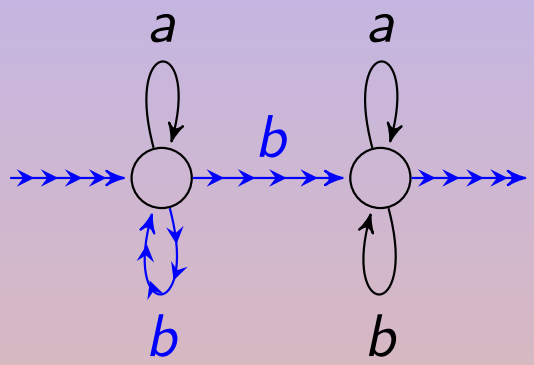
Typing Automata and Rational Expressions

From Recognizers to Weighted Automata



bb

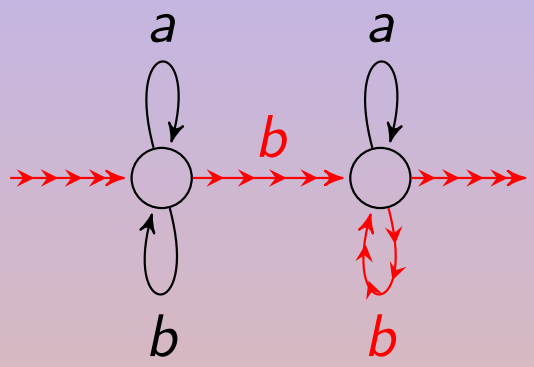
From Recognizers to Weighted Automata



$\top \wedge \top$

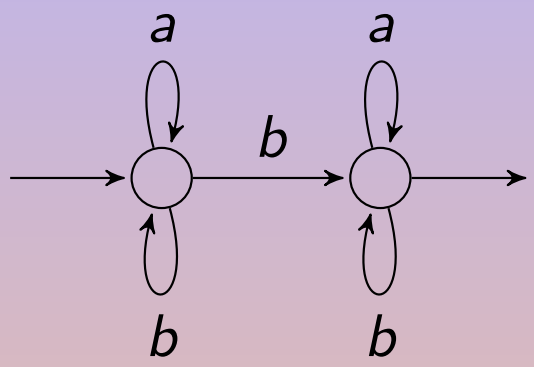
bb

From Recognizers to Weighted Automata



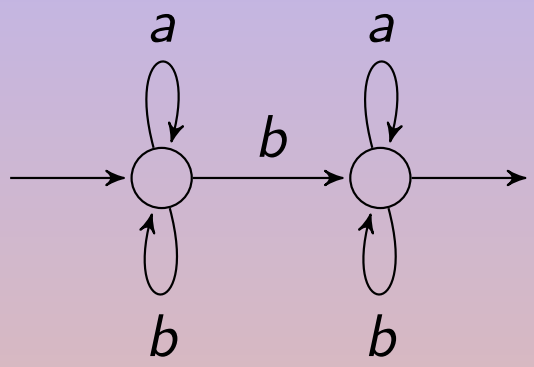
$\top \wedge \top \quad \top \wedge \top \qquad\qquad bb$

From Recognizers to Weighted Automata



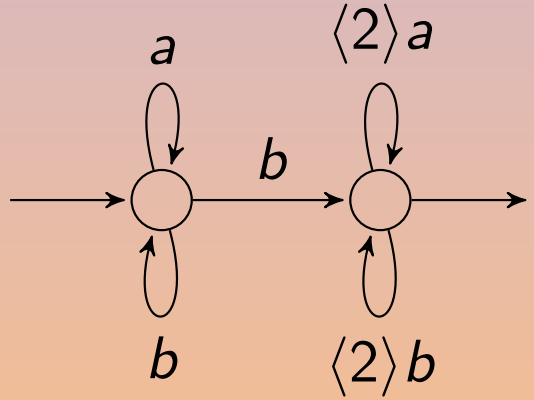
$$\top \wedge \top \vee \top \wedge \top = \top \qquad\qquad bb \to \top$$

From Recognizers to Weighted Automata



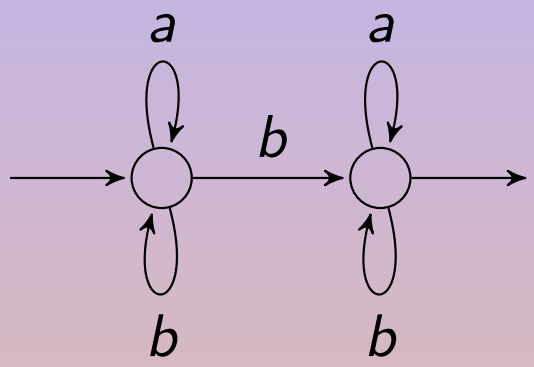
$$\top \wedge \top \vee \top \wedge \top = \top$$

$$bb \to \top$$

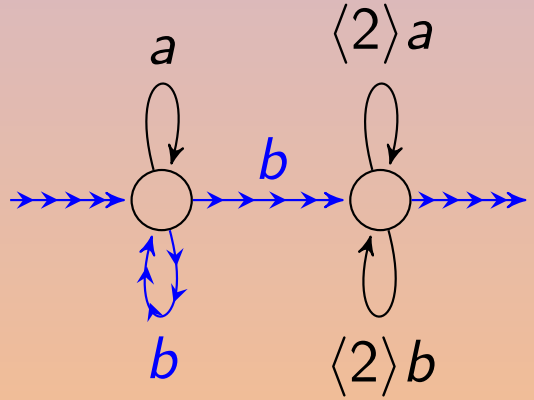


$$bb$$

From Recognizers to Weighted Automata

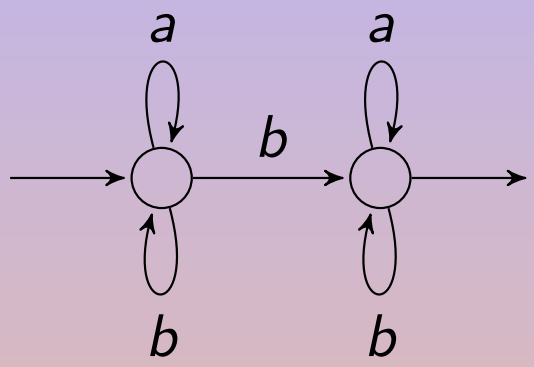


$$\top \wedge \top \vee \top \wedge \top = \top \qquad bb \to \top$$

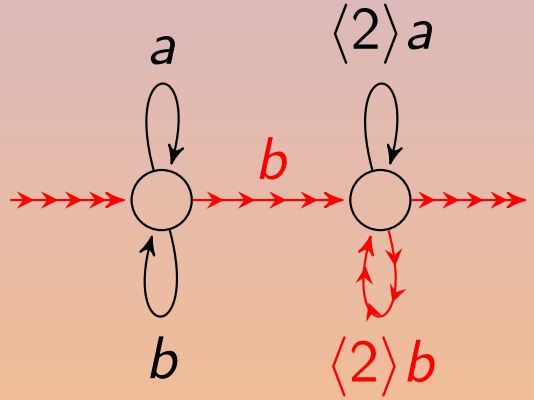


$$1 \times 1 \qquad bb$$

From Recognizers to Weighted Automata

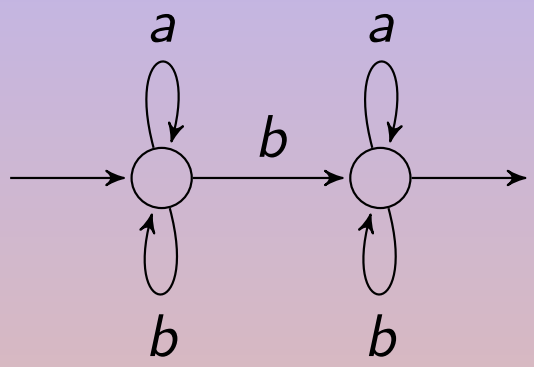


$$\top \wedge \top \vee \top \wedge \top = \top \qquad bb \to \top$$

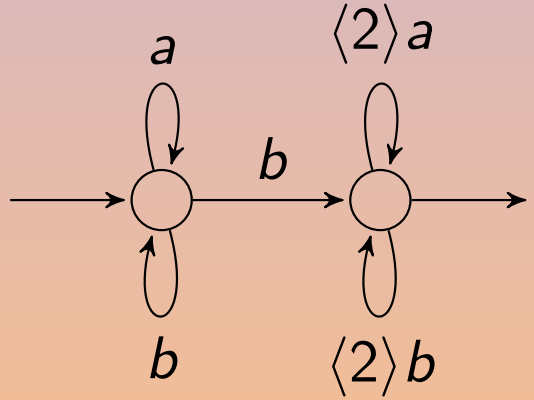


$$1 \times 1 + 2 \times 1 \qquad bb$$

From Recognizers to Weighted Automata



$$\top \wedge \top \vee \top \wedge \top = \top \qquad bb \to \top$$

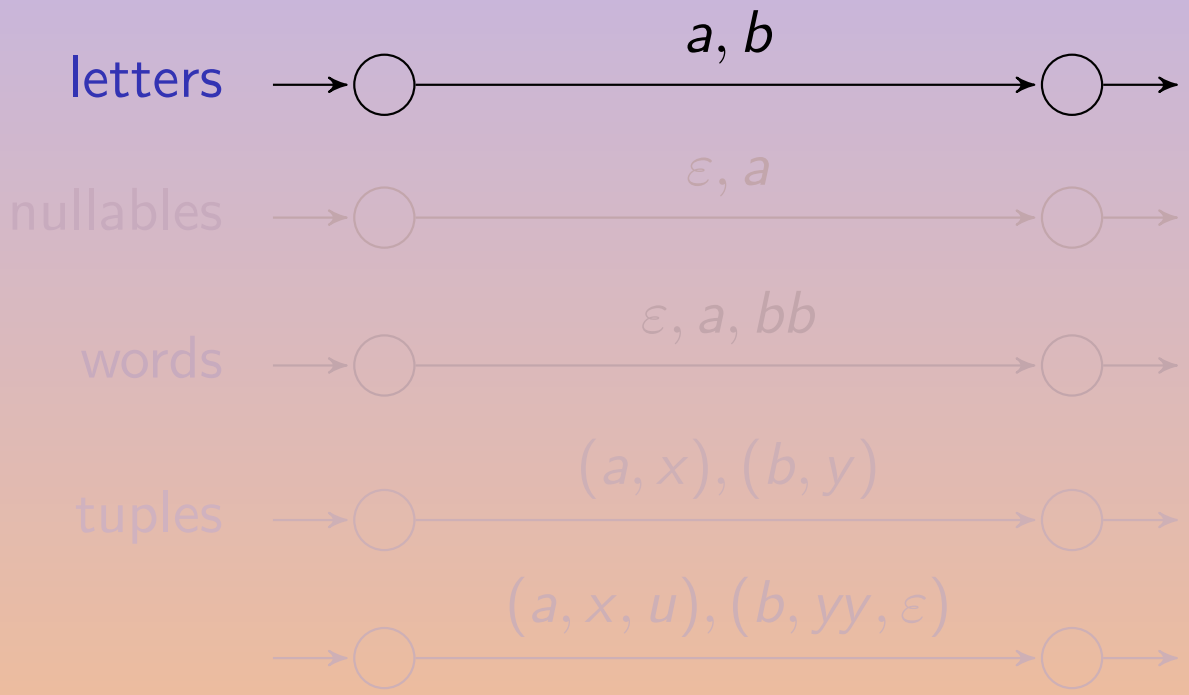


$$1 \times 1 + 2 \times 1 = 3 \qquad bb \to 3$$

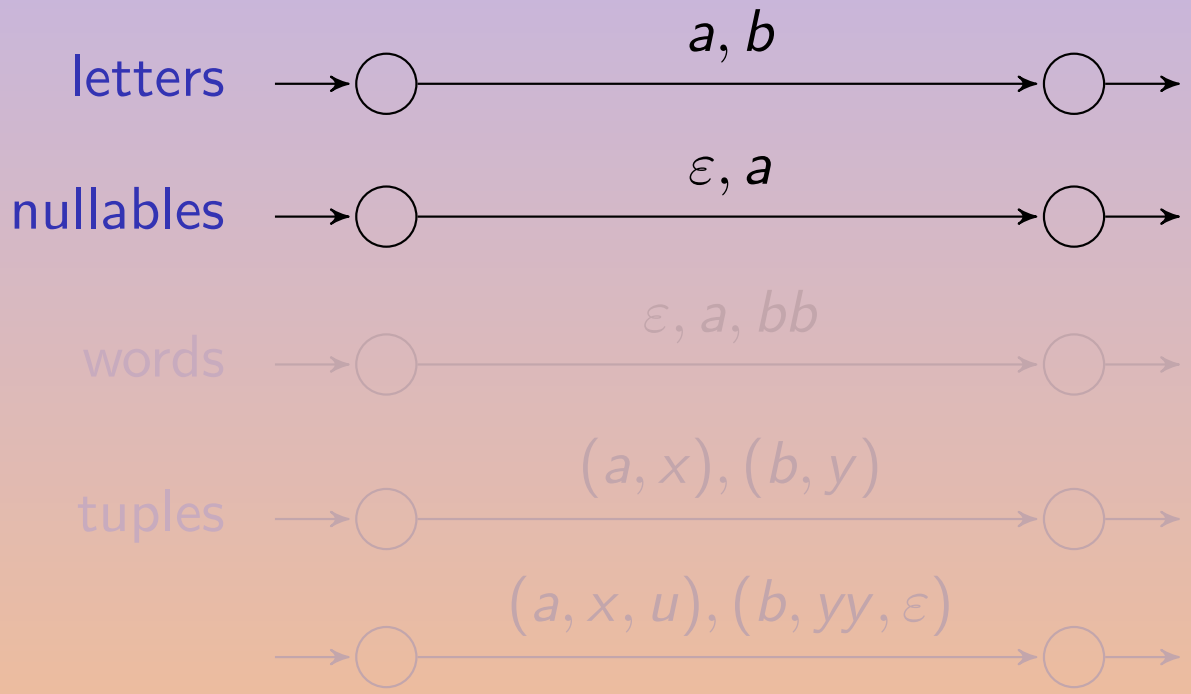
Many Different Kinds of Weights

- $\langle \mathbb{B}, \vee, \wedge \rangle$
- $\langle \mathbb{Z}, +, \times \rangle$
- $\langle \mathbb{Q}, +, \times \rangle$
- $\langle \mathbb{R}, +, \times \rangle$
- $\langle \mathbb{Z}, \mathsf{min}, + \rangle$
- Rational expressions
- Tuples
- ...

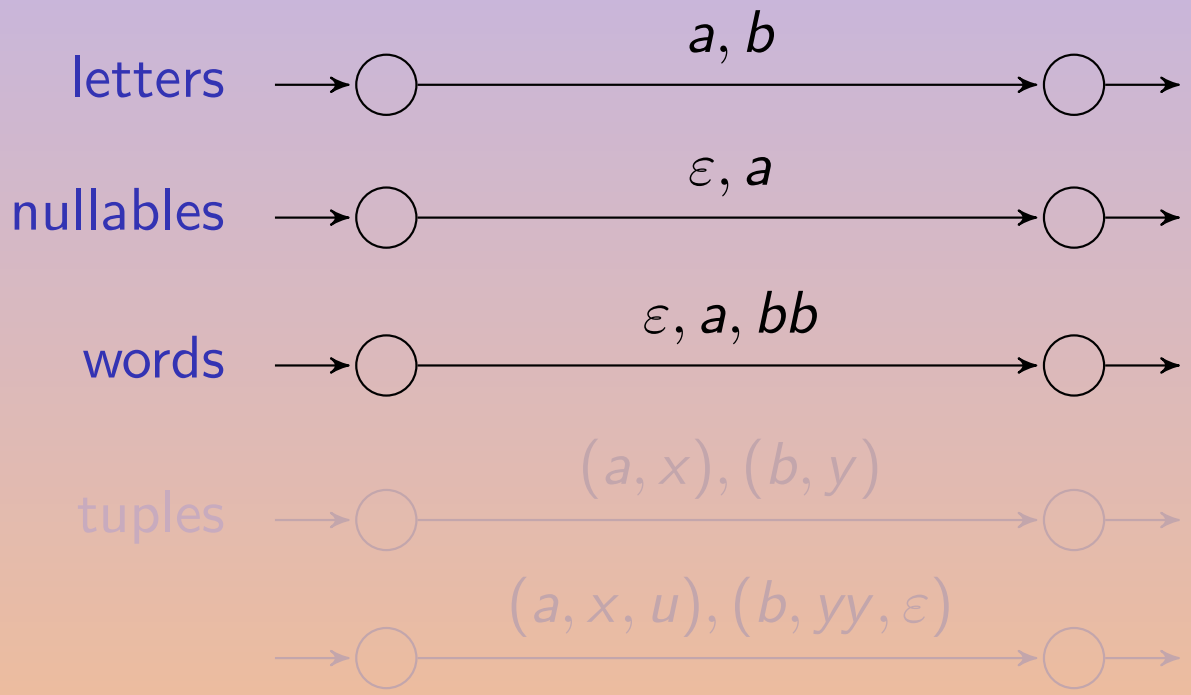
Many Different Kinds of Labels



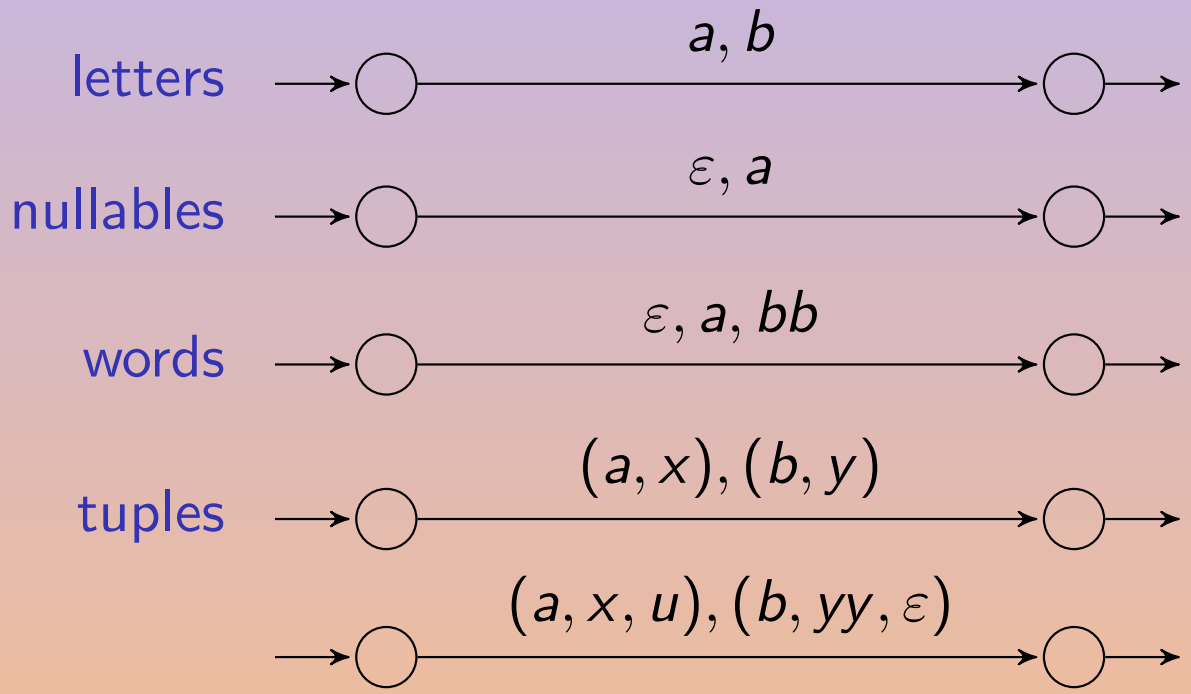
Many Different Kinds of Labels



Many Different Kinds of Labels



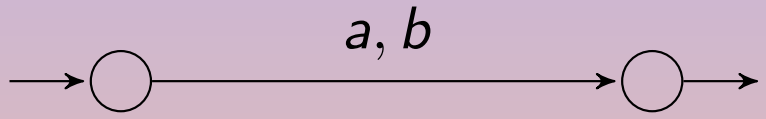
Many Different Kinds of Labels



Contexts

$$\mathit{context} ::= \mathit{labelset} \to \mathit{weightset}$$

$\{a, b\} \to \mathbb{B}$

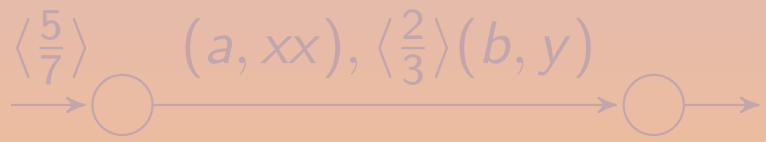


$(\varepsilon + a + b)^*$

$\{a, b, c\}^? \to \mathbb{Z}$

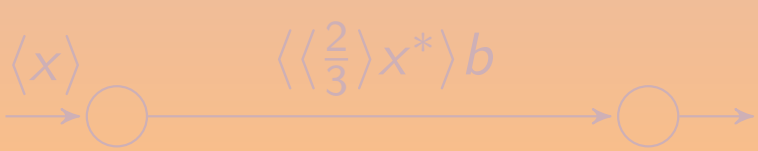
$\langle 2\rangle(\langle -2\rangle a + b)^*$

$\{a, b, c\} \times \{x, y, z\}^* \to \mathbb{Q}$



$\langle \frac{1}{2}\rangle(a, x) + \langle \frac{1}{3}\rangle(b, y)^*$

$\{a, b\} \to \mathsf{RatE}[\{x, y\} \to \mathbb{Q}]$

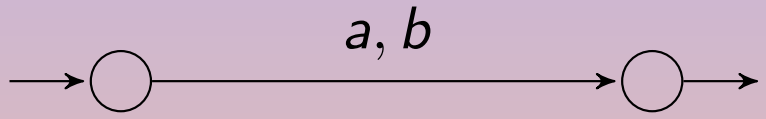


$\left(\langle x\rangle a + \langle\langle \frac{1}{3}\rangle(x + y)^*\rangle b\right)^*$

Contexts

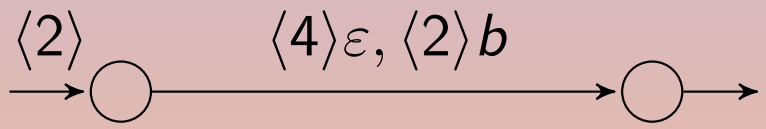
$$\mathit{context} ::= \mathit{labelset} \to \mathit{weightset}$$

$\{a, b\} \to \mathbb{B}$



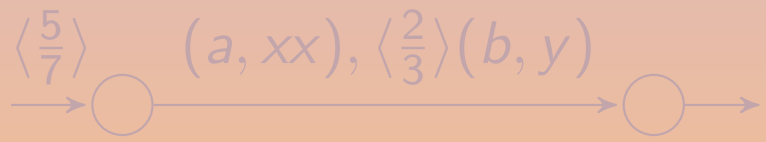
$(\varepsilon + a + b)^*$

$\{a, b, c\}^? \to \mathbb{Z}$



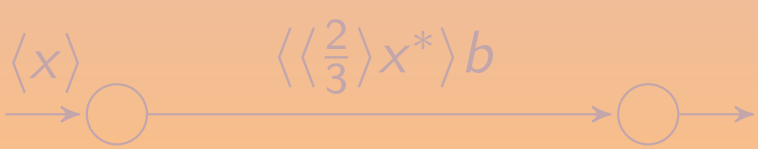
$\langle 2\rangle(\langle -2\rangle a + b)^*$

$\{a, b, c\} \times \{x, y, z\}^* \to \mathbb{Q}$



$\langle \frac{1}{2}\rangle(a, x) + \langle \frac{1}{3}\rangle(b, y)^*$

$\{a, b\} \to \mathrm{RatE}[\{x, y\} \to \mathbb{Q}]$

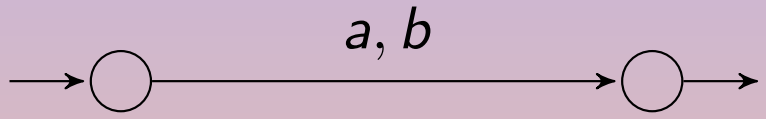


$\left(\langle x\rangle a + \langle\langle \frac{1}{3}\rangle(x + y)^*\rangle b\right)^*$

Contexts

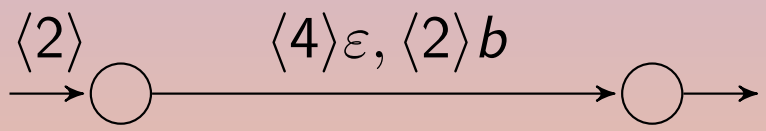
$$context ::= labelset \to weightset$$

$\{a, b\} \to \mathbb{B}$



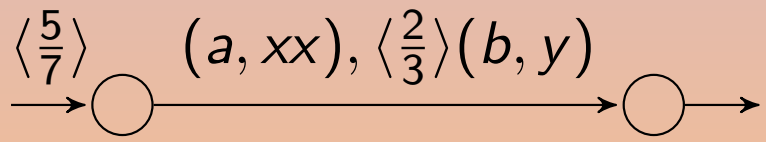
$(\varepsilon + a + b)^*$

$\{a, b, c\}^? \to \mathbb{Z}$



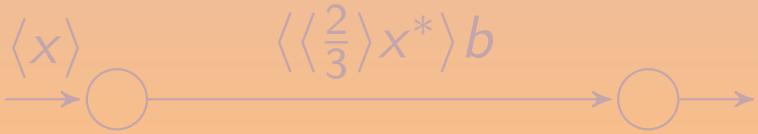
$\langle 2\rangle(\langle -2\rangle a + b)^*$

$\{a, b, c\} \times \{x, y, z\}^* \to \mathbb{Q}$



$\langle \frac{1}{2}\rangle(a, x) + \langle \frac{1}{3}\rangle(b, y)^*$

$\{a, b\} \to \mathrm{RatE}[\{x, y\} \to \mathbb{Q}]$

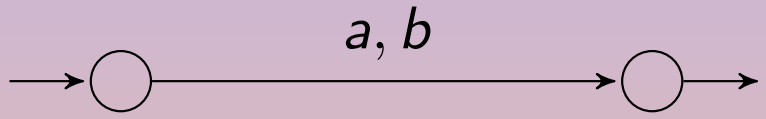


$\left(\langle x\rangle a + \langle\langle \frac{1}{3}\rangle(x + y)^*\rangle b\right)^*$

Contexts

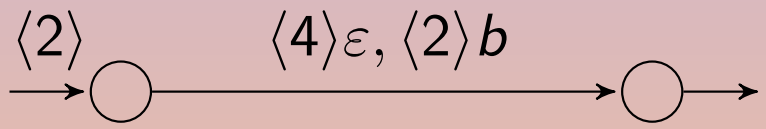
$$\mathit{context} ::= \mathit{labelset} \to \mathit{weightset}$$

$\{a, b\} \to \mathbb{B}$



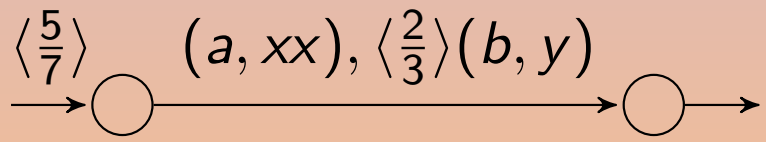
$$(\varepsilon + a + b)^*$$

$\{a, b, c\}^? \to \mathbb{Z}$



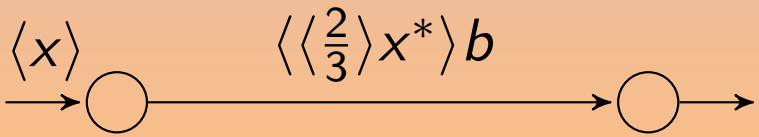
$$\langle 2\rangle(\langle -2\rangle a + b)^*$$

$\{a, b, c\} \times \{x, y, z\}^* \to \mathbb{Q}$



$$\langle \tfrac{1}{2}\rangle(a, x) + \langle \tfrac{1}{3}\rangle(b, y)^*$$

$\{a, b\} \to \mathrm{RatE}[\{x, y\} \to \mathbb{Q}]$



$$\left(\langle x\rangle a + \langle\langle \tfrac{1}{3}\rangle(x + y)^*\rangle b\right)^*$$

A Calculus on Types

Subtyping

$$A <: B$$

All values of type A are also values of type B

roughly, "$A \subseteq B$"

Labelsets and Weightsets Subtypes

$\{\varepsilon\} <: A^?$ $\quad A <: A^?$ $\quad A^? <: A^*$

Labelsets and Weightsets Subtypes

$\{\varepsilon\} <: A^?$ $\quad A <: A^?$ $\quad A^? <: A^*$

With A, B alphabets such that $A \subseteq B$:

$A <: B$ $\quad A^? <: B^?$ $\quad A^* <: B^*$

Labelsets and Weightsets Subtypes

$\{\varepsilon\} <: A^?$ $\quad A <: A^?$ $\quad A^? <: A^*$

$\mathbb{B} <: \mathbb{N} <: \mathbb{Z} <: \mathbb{Q} <: \mathbb{R}$

$\mathbb{B} <: \mathbb{Z}_{\mathsf{min}}$

With A, B alphabets such that $A \subseteq B$:

$A <: B$ $\quad A^? <: B^?$ $\quad A^* <: B^*$

Labelsets and Weightsets Subtypes

$\{\varepsilon\} <: A^?$ $\quad A <: A^?$ $\quad A^? <: A^*$

$\mathbb{B} <: \mathbb{N} <: \mathbb{Z} <: \mathbb{Q} <: \mathbb{R}$

$\mathbb{B} <: \mathbb{Z}_{\mathsf{min}}$

With A, B alphabets such that $A \subseteq B$:

$A <: B$ $\quad A^? <: B^?$ $\quad A^* <: B^*$

$L <: \mathsf{RatE}[L \to W]$ $\qquad$ $W <: \mathsf{RatE}[L \to W]$

Subtype On Contexts, Expressions and Automata

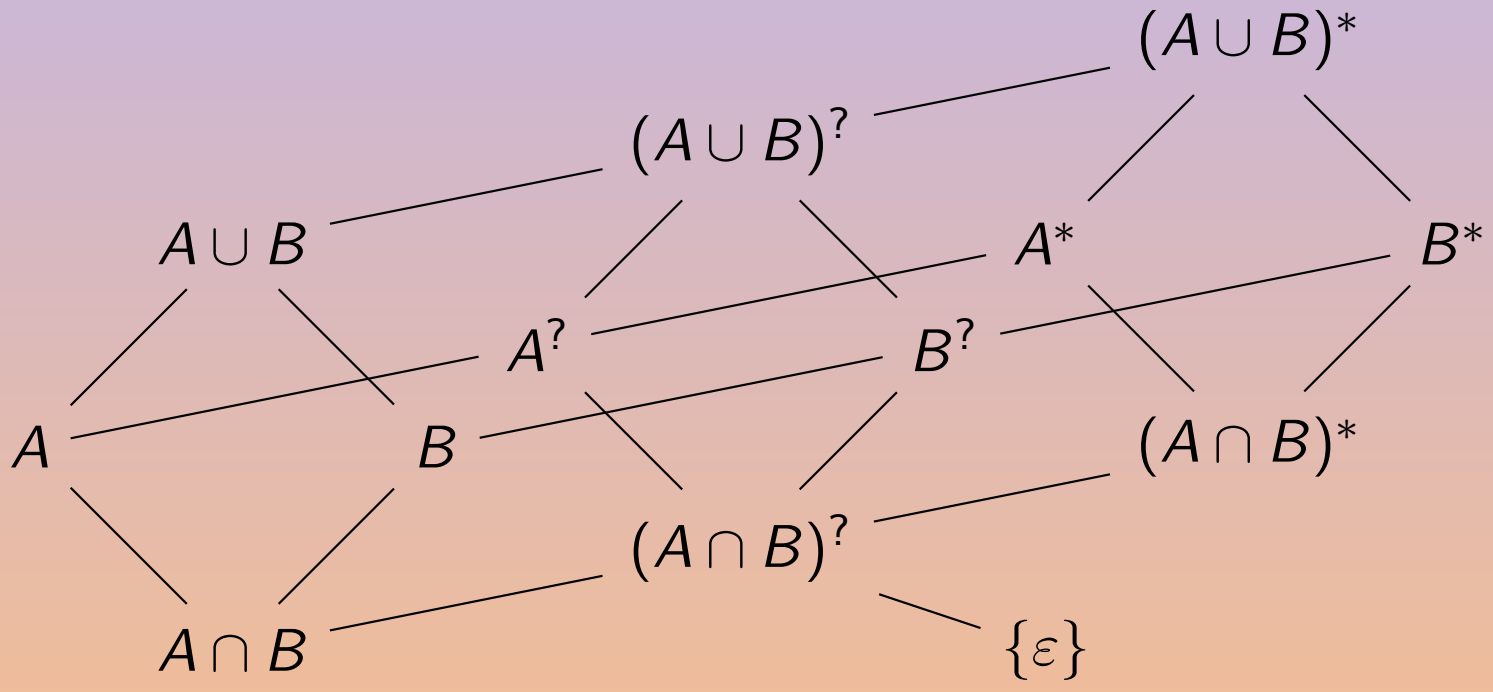
$$\frac{L_1 <: L_2 \qquad W_1 <: W_2}{(L_1 \to W_1) <: (L_2 \to W_2)}$$

$$\frac{C_1 <: C_2}{\mathsf{RatE}[C_1] <: \mathsf{RatE}[C_2]}$$

$$\frac{C_1 <: C_2}{\mathsf{Aut}[C_1] <: \mathsf{Aut}[C_2]}$$

Labelset Subtypes

Let A, B be alphabets:



Meet and Join

$V_1 \vee V_2$ (join)
: The least upper bound between V_1 and V_2.

$V_1 \wedge V_2$ (meet)
: The greatest lower bound between V_1 and V_2.

$$\mathbb{Q} \quad \vee \quad \mathrm{RatE}[\{x, y, z\} \to \mathbb{B}] \quad = \quad \mathrm{RatE}[\{x, y, z\} \to \mathbb{Q}]$$

$$\{a, b, c\} \quad \wedge \quad \{a, b, d\} \quad = \quad \{a, b\}$$

Meet and Join

$V_1 \vee V_2$ (join)
The least upper bound between V_1 and V_2.

$V_1 \wedge V_2$ (meet)
The greatest lower bound between V_1 and V_2.

$$\mathbb{Q} \quad \vee \quad \mathsf{RatE}[\{x, y, z\} \to \mathbb{B}] \quad = \quad \mathsf{RatE}[\{x, y, z\} \to \mathbb{Q}]$$

$$\{a, b, c\} \quad \wedge \quad \{a, b, d\} \quad = \quad \{a, b\}$$

Meet and Join

$V_1 \vee V_2$ (join)
The least upper bound between V_1 and V_2.

$V_1 \wedge V_2$ (meet)
The greatest lower bound between V_1 and V_2.

$$\mathbb{Q} \vee \mathsf{RatE}[\{x, y, z\} \to \mathbb{B}] = \mathsf{RatE}[\{x, y, z\} \to \mathbb{Q}]$$

$$\{a, b, c\} \wedge \{a, b, d\} = \{a, b\}$$

Union of Automata

$$\frac{\mathcal{A}_1 : L_1 \to W_1 \qquad \mathcal{A}_2 : L_2 \to W_2}{\mathcal{A}_1 \cup \mathcal{A}_2 : L_1 \vee L_2 \to W_1 \vee W_2}$$

Union of Automata

$$\frac{\mathcal{A}_1 : L_1 \to W_1 \qquad \mathcal{A}_2 : L_2 \to W_2}{\mathcal{A}_1 \cup \mathcal{A}_2 : L_1 \vee L_2 \to W_1 \vee W_2}$$

$$\frac{\mathcal{A}_1 : \{a, b, c\} \to \mathbb{Q} \qquad \mathcal{A}_2 : \{a, b, d\} \to \mathsf{RatE}[\{x, y, z\} \to \mathbb{B}]}{\mathcal{A}_1 \cup \mathcal{A}_2 : \qquad \to}$$

Union of Automata

$$\frac{\mathcal{A}_1 : L_1 \to W_1 \qquad \mathcal{A}_2 : L_2 \to W_2}{\mathcal{A}_1 \cup \mathcal{A}_2 : L_1 \vee L_2 \to W_1 \vee W_2}$$

$$\frac{\mathcal{A}_1 : \{a, b, c\} \to \mathbb{Q} \qquad \mathcal{A}_2 : \{a, b, d\} \to \mathsf{RatE}[\{x, y, z\} \to \mathbb{B}]}{\mathcal{A}_1 \cup \mathcal{A}_2 : \qquad \to}$$

Union of Automata

$$\frac{\mathcal{A}_1 : L_1 \to W_1 \qquad \mathcal{A}_2 : L_2 \to W_2}{\mathcal{A}_1 \cup \mathcal{A}_2 : L_1 \vee L_2 \to W_1 \vee W_2}$$

$$\frac{\mathcal{A}_1 : \{a, b, c\} \to \mathbb{Q} \qquad \mathcal{A}_2 : \{a, b, d\} \to \mathsf{RatE}[\{x, y, z\} \to \mathbb{B}]}{\mathcal{A}_1 \cup \mathcal{A}_2 : \{a, b, c, d\} \to}$$

because $\{a, b, c\} \vee \{a, b, d\} = \{a, b, c, d\}$

Union of Automata

$$\frac{\mathcal{A}_1 : L_1 \to W_1 \qquad \mathcal{A}_2 : L_2 \to W_2}{\mathcal{A}_1 \cup \mathcal{A}_2 : L_1 \vee L_2 \to W_1 \vee W_2}$$

$$\frac{\mathcal{A}_1 : \{a, b, c\} \to \mathbb{Q} \qquad \mathcal{A}_2 : \{a, b, d\} \to \mathsf{RatE}[\{x, y, z\} \to \mathbb{B}]}{\mathcal{A}_1 \cup \mathcal{A}_2 : \{a, b, c, d\} \to}$$

because $\{a, b, c\} \vee \{a, b, d\} = \{a, b, c, d\}$

Union of Automata

$$\frac{\mathcal{A}_1 : L_1 \to W_1 \qquad \mathcal{A}_2 : L_2 \to W_2}{\mathcal{A}_1 \cup \mathcal{A}_2 : L_1 \vee L_2 \to W_1 \vee W_2}$$

$$\frac{\mathcal{A}_1 : \{a, b, c\} \to \mathbb{Q} \qquad \mathcal{A}_2 : \{a, b, d\} \to \mathsf{RatE}[\{x, y, z\} \to \mathbb{B}]}{\mathcal{A}_1 \cup \mathcal{A}_2 : \{a, b, c, d\} \to \mathsf{RatE}[\{x, y, z\} \to \mathbb{Q}]}$$

$$\textit{because} \quad \{a, b, c\} \vee \{a, b, d\} = \{a, b, c, d\}$$

$$\textit{because} \quad \mathbb{Q} \vee \mathsf{RatE}[\{x, y, z\} \to \mathbb{B}] = \mathsf{RatE}[\{x, y, z\} \to \mathbb{Q}]$$

Synchronized Product of Automata

$$\frac{\mathcal{A}_1 : A_1 \to W_1 \qquad \mathcal{A}_2 : A_2 \to W_2}{\mathcal{A}_1 \,\&\, \mathcal{A}_2 :}$$

Synchronized Product of Automata

$$\frac{\mathcal{A}_1 : A_1 \to W_1 \qquad \mathcal{A}_2 : A_2 \to W_2}{\mathcal{A}_1 \,\&\, \mathcal{A}_2 : A_1 \wedge A_2 \to W_1 \vee W_2}$$

Synchronized Product of Automata

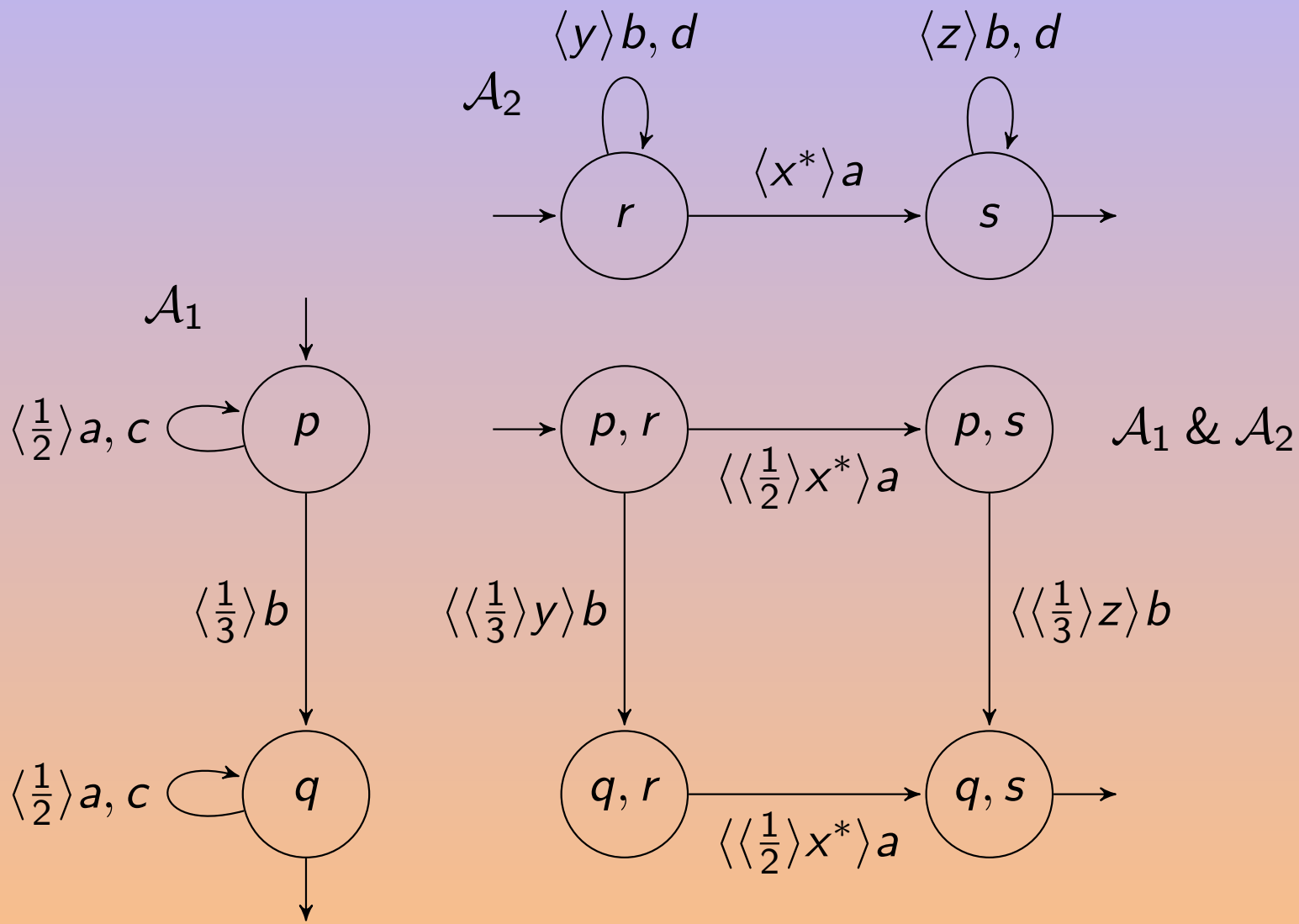
$$\frac{\mathcal{A}_1 : A_1 \to W_1 \qquad \mathcal{A}_2 : A_2 \to W_2}{\mathcal{A}_1 \,\&\, \mathcal{A}_2 : A_1 \wedge A_2 \to W_1 \vee W_2}$$

$$\frac{\mathcal{A}_1 : \{a, b, c\} \to \mathbb{Q} \qquad \mathcal{A}_2 : \{a, b, d\} \to \mathsf{RatE}[\{x, y, z\} \to \mathbb{B}]}{\mathcal{A}_1 \,\&\, \mathcal{A}_2 : \{a, b\} \to \mathsf{RatE}[\{x, y, z\} \to \mathbb{Q}]}$$

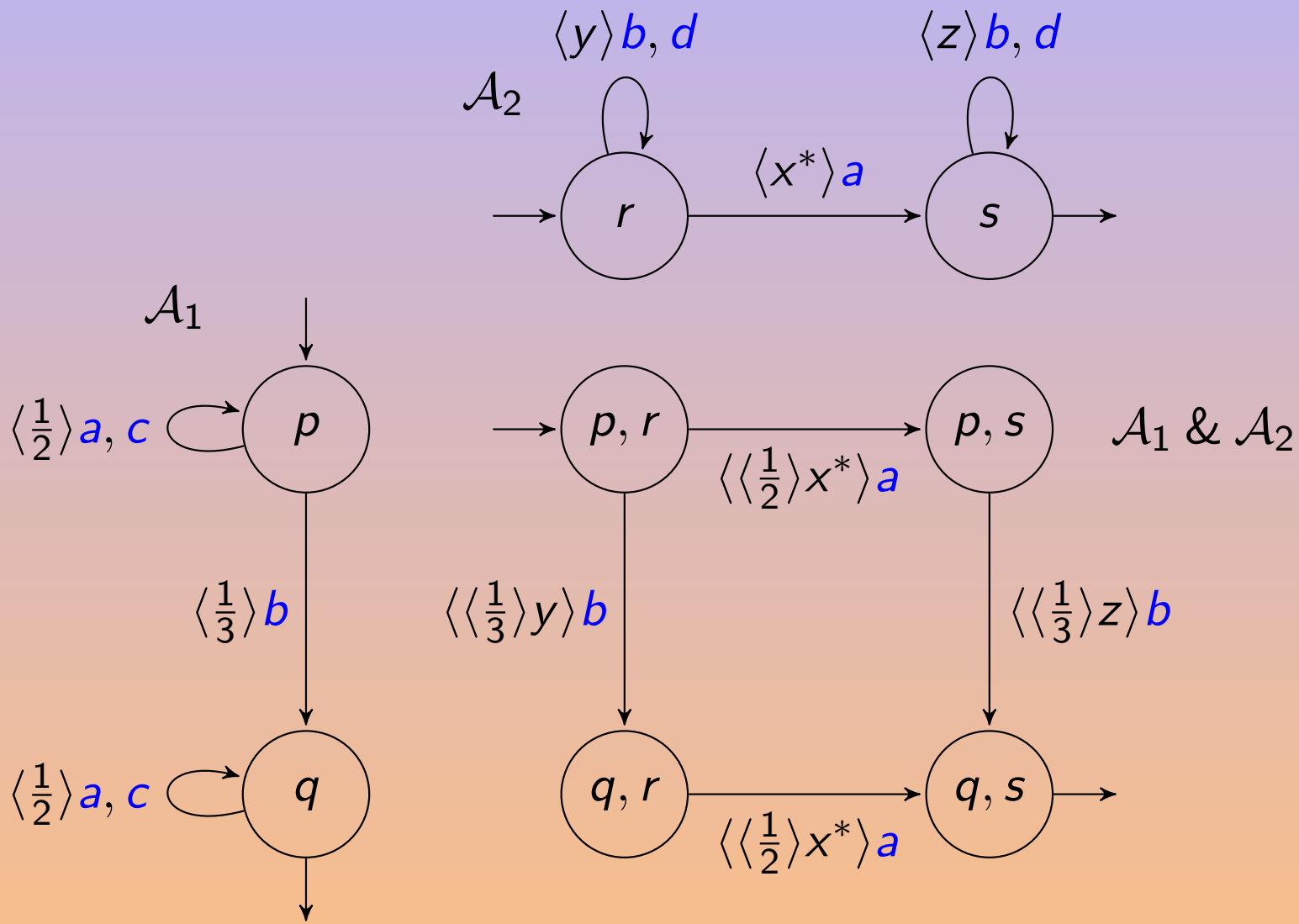
because $\quad \{a, b, c\} \wedge \{a, b, d\} = \{a, b\}$

because $\quad \mathbb{Q} \vee \mathsf{RatE}[\{x, y, z\} \to \mathbb{B}] = \mathsf{RatE}[\{x, y, z\} \to \mathbb{Q}]$

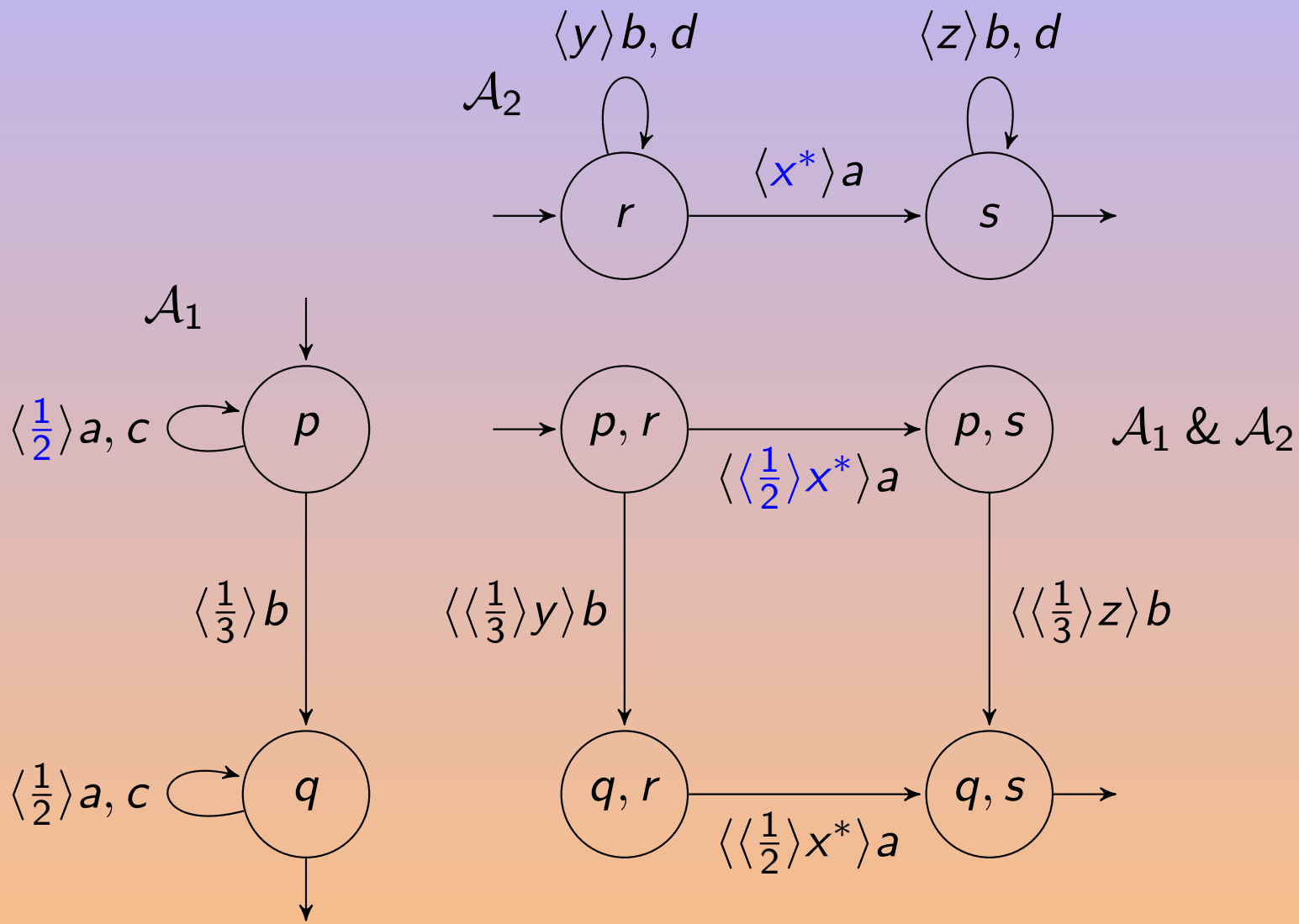
Synchronized Product in Vaucanson



Synchronized Product in Vaucanson



Synchronized Product in Vaucanson



Use in Vaucanson

Use of Types

- Type objects exist, as values, at a low level
- Dynamic objects wrap low-level objects
- The dynamic library resolves calls discriminating on type objects
- The dispatching routine computes resulting types
- Types can[1] be used to generate code at run time

[1] After publication: now they are.

Use of Types

- Type objects exist, as values, at a low level
- Dynamic objects wrap low-level objects
- The dynamic library resolves calls discriminating on type objects
- The dispatching routine computes resulting types
- Types can[1] be used to generate code at run time

[1] After publication: now they are.

Use of Types

- Type objects exist, as values, at a low level
- Dynamic objects wrap low-level objects
- The dynamic library resolves calls discriminating on type objects
- The dispatching routine computes resulting types
- Types can[1] be used to generate code at run time

[1] After publication: now they are.

Operations on Rational Expressions

$$\frac{\mathsf{E}_1 : \mathsf{RatE}[L_1 \to W_1] \qquad \mathsf{E}_2 : \mathsf{RatE}[L_2 \to W_2]}{\mathsf{E}_1 \odot \mathsf{E}_2 : \mathsf{RatE}[L_1 \vee L_2 \to W_1 \vee W_2]} \odot \in \{\cdot, +\}$$

$$\frac{w_1 : W_1 \qquad \mathsf{E}_2 : \mathsf{RatE}[L_2 \to W_2]}{w_1 \cdot \mathsf{E}_2 : \mathsf{RatE}[L_2 \to W_1 \vee W_2]}$$

$$\frac{\mathsf{E}_1 : \mathsf{RatE}[L_1 \to W_1] \qquad w_2 : W_2}{\mathsf{E}_1 \cdot w_2 : \mathsf{RatE}[L_1 \to W_1 \vee W_2]}$$

Operations on Automata

$$\frac{\mathcal{A}_1 : L_1 \to W_1 \qquad \mathcal{A}_2 : L_2 \to W_2}{\mathcal{A}_1 \odot \mathcal{A}_2 : L_1 \vee L_2 \to W_1 \vee W_2} \odot \in \{\cdot, +\}$$

$$\frac{\mathcal{A}_1 : A_1 \to W_1 \qquad \mathcal{A}_2 : A_2 \to W_2}{\mathcal{A}_1 \odot \mathcal{A}_2 : A_1 \vee A_2 \to W_1 \vee W_2} \odot \in \{(\!(, \uparrow\}$$

$$\frac{w_1 : W_1 \qquad \mathcal{A}_2 : L_2 \to W_2}{w_1 \cdot \mathcal{A}_2 : L_2 \to W_1 \vee W_2}$$

$$\frac{\mathcal{A}_1 : L_1 \to W_1 \qquad w_2 : W_2}{\mathcal{A}_1 \cdot w_2 : L_1 \to W_1 \vee W_2}$$

Conclusion

"State"

- Acceptors, transducers
- Efficient static API
- Flexible dynamic API with runtime compilation
- User[/student]-friendly IPython interface

"Transition"

- Better transducer support
- Improved type-checking errors
- Richer expressions
- Metadata on states[2]
- ...

[2] After publication: now largely done.

Conclusion

"State"

- Acceptors, transducers
- Efficient static API
- Flexible dynamic API with runtime compilation
- User[/student]-friendly IPython interface

"Transition"

- Better transducer support
- Improved type-checking errors
- Richer expressions
- Metadata on states[2]
- ...

[2] After publication: now largely done.

Questions?