Kronecker Algebra

Computer Systems

Johann Blieberger

1. Kronecker Product
2. Verifying Programs
3. The Kronecker Sum

Kronecker Product

Kronecker Product of Finite State Machines (FSMs)

Definition

Given an m-by-n matrix A and a p-by-q matrix B, their Kronecker product denoted by $A \otimes B$ is a mp-by-nq block matrix defined by

$$A \otimes B = \begin{pmatrix} a_{1,1}B & \dots & a_{1,n}B \\ \vdots & \ddots & \vdots \\ a_{m,1}B & \dots & a_{m,n}B \end{pmatrix}.$$

Kronecker Product

Example

For example, if

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix}$$

and

$$B = \begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix},$$

then

$$A \otimes B = \begin{pmatrix} a_{1,1}b_{1,1} & a_{1,1}b_{1,2} & a_{1,1}b_{1,3} & a_{1,2}b_{1,1} & a_{1,2}b_{1,2} & a_{1,2}b_{1,3} \\ a_{1,1}b_{2,1} & a_{1,1}b_{2,2} & a_{1,1}b_{2,3} & a_{1,2}b_{2,1} & a_{1,2}b_{2,2} & a_{1,2}b_{2,3} \\ a_{1,1}b_{3,1} & a_{1,1}b_{3,2} & a_{1,1}b_{3,3} & a_{1,2}b_{3,1} & a_{1,2}b_{3,2} & a_{1,2}b_{3,3} \\ a_{2,1}b_{1,1} & a_{2,1}b_{1,2} & a_{2,1}b_{1,3} & a_{2,2}b_{1,1} & a_{2,2}b_{1,2} & a_{2,2}b_{1,3} \\ a_{2,1}b_{2,1} & a_{2,1}b_{2,2} & a_{2,1}b_{2,3} & a_{2,2}b_{2,1} & a_{2,2}b_{2,2} & a_{2,2}b_{2,3} \\ a_{2,1}b_{3,1} & a_{2,1}b_{3,2} & a_{2,1}b_{3,3} & a_{2,2}b_{3,1} & a_{2,2}b_{3,2} & a_{2,2}b_{3,3} \end{pmatrix}.$$

Kronecker Product

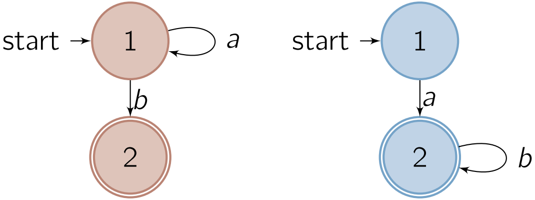


Abbildung: FSMs A (left) and B (right)

Kronecker Product

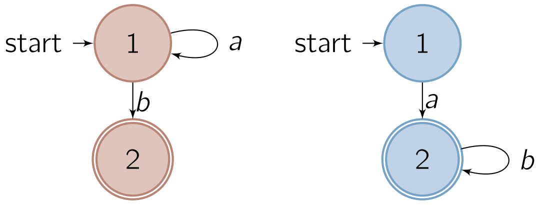


Abbildung: FSMs A (left) and B (right)

Example

matrices

$$A = \begin{pmatrix} a & b \\ 0 & 0 \end{pmatrix}, \; B = \begin{pmatrix} 0 & a \\ 0 & b \end{pmatrix}, \; A \otimes B = \begin{pmatrix} \cdot & aa & \cdot & ba \\ \cdot & ab & \cdot & bb \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

Kronecker Product

Initial and Final States

Let S_A and S_B be the initial state vectors of the operands, F_A and F_B their final state vectors. Then the initial vector of the Kronecker product is given by $S_A \otimes S_B$ and its final vector is $F_A \otimes F_B$.

Example

$S_A = (1, 0)$, $S_B = (1, 0)$, $F_A = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, and $F_B = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. Thus $S_{A \otimes B} = (1, 0, 0, 0)$ and

$$F_{A \otimes B} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

which simply state that the initial state of the Kronecker product FSM $A \otimes B$ is state 1 and its final state is state 4.

Kronecker Product for FSMs

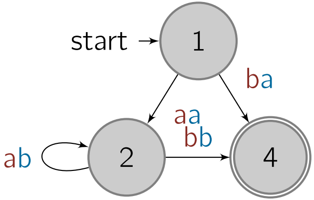


Abbildung: Graphical Representation of $A \otimes B$

Kronecker Product for FSMs

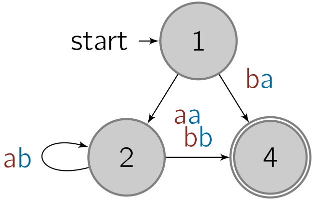


Abbildung: Graphical Representation of $A \otimes B$

Surprise

matrix has size four

Kronecker Product for FSMs

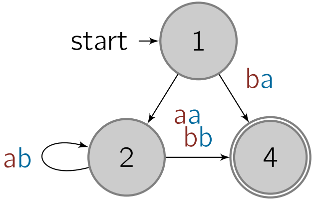


Abbildung: Graphical Representation of $A \otimes B$

Surprise

matrix has size four

only three states above

Kronecker Product for FSMs

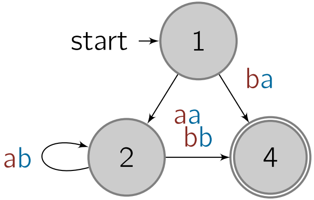


Abbildung: Graphical Representation of $A \otimes B$

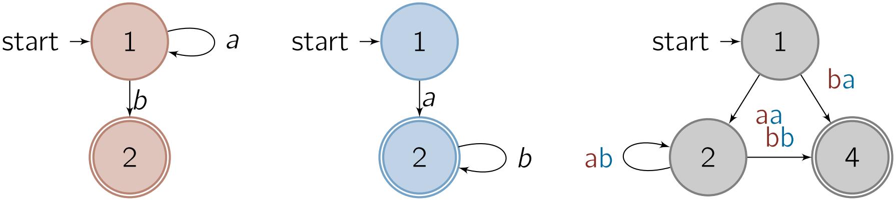
Surprise

matrix has size four

only three states above

State 3 cannot be reached from the initial state 1!

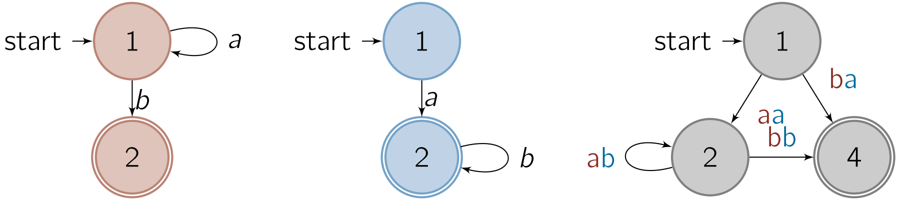
Kronecker Product for FSMs



Both FSMs, *A* and *B* perform their state transitions in *lockstep*.

1. At the beginning, both FSMs are in their initial state. *B* has to proceed to state 2, thereby generating output *a*. *A* has two possible successor states:
   - A. *A* stays in state 1, producing output *a*. This corresponds to state transition $1 \rightarrow 2$ in the product FSM.
   - B. *A* proceeds to its state 2 (output: *b*). This corresponds to state transition $1 \rightarrow 4$ in the product FSM.

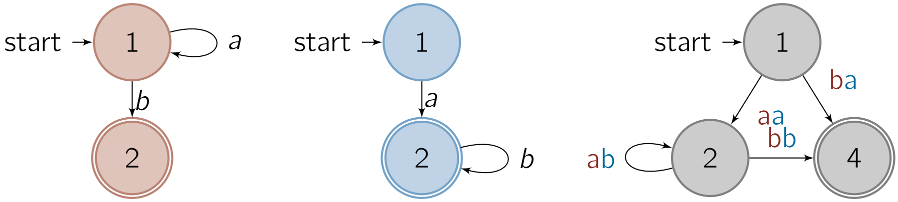
Kronecker Product for FSMs



Both FSMs, *A* and *B* perform their state transitions in *lockstep*.

2 If the previous step was 1A, both FSMs can now produce together an arbitrary number of *ab*-pairs. This corresponds to the transition $2 \rightarrow 2$ in the product FSM. When *A* issues a *b*, *B* also has to produce a *b*. This corresponds to transition $2 \rightarrow 4$ in the product FSM. Then both FSMs are in their final states. *A* cannot do a further state transition and both FSMs and the product FSM terminate.

Kronecker Product for FSMs



Both FSMs, *A* and *B* perform their state transitions in *lockstep*.

3 If the previous step was 1B, both FSMs are in their final states. Since *A* cannot proceed, both FSMs and the product FSM terminate.

Kronecker Product for FSMs

Theorem

Given two FSMs A and B, their Kronecker product FSM generates the same output than A and B do when they execute in lockstep. □

Kronecker Product for FSMs

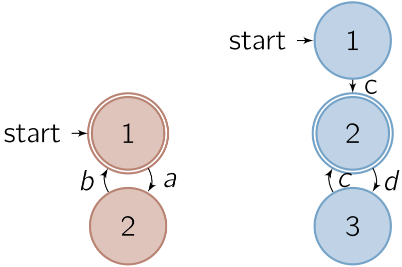


Abbildung: FSMs C (left) and D (right)

Kronecker Product for FSMs

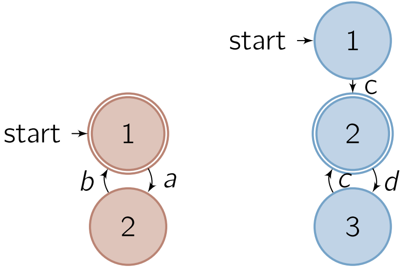


Abbildung: FSMs C (left) and D (right)

C terminates only when it has done an even number of state transitions

Kronecker Product for FSMs

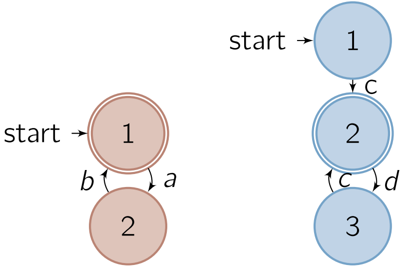


Abbildung: FSMs C (left) and D (right)

C terminates only when it has done an even number of state transitions

D can terminate only after it has done an odd number of transitions

Kronecker Product for FSMs

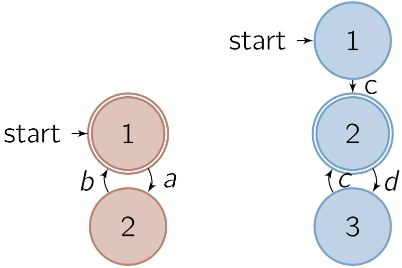


Abbildung: FSMs C (left) and D (right)

C terminates only when it has done an even number of state transitions
D can terminate only after it has done an odd number of transitions
overall output of C ... $(ab)^*$

Kronecker Product for FSMs

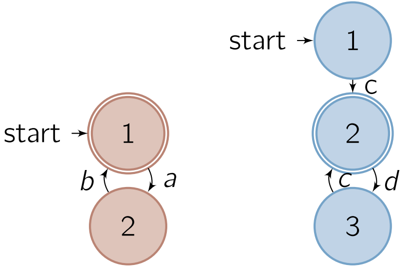


Abbildung: FSMs C (left) and D (right)

- C terminates only when it has done an even number of state transitions
- D can terminate only after it has done an odd number of transitions
- overall output of C ... $(ab)^*$
- overall output of D ... $c(dc)^*$

Kronecker Product for FSMs

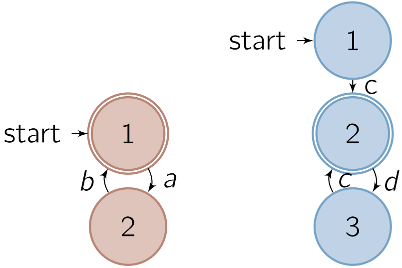


Abbildung: FSMs C (left) and D (right)

C terminates only when it has done an even number of state transitions
D can terminate only after it has done an odd number of transitions
overall output of C … $(ab)^*$
overall output of D … $c(dc)^*$
the Kronecker product of C and D …

Kronecker Product for FSMs

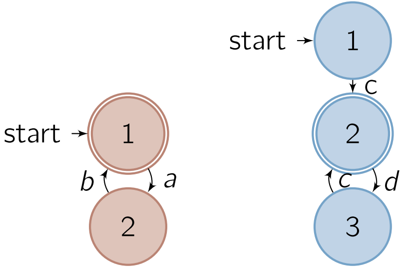


Abbildung: FSMs C (left) and D (right)

C terminates only when it has done an even number of state transitions
D can terminate only after it has done an odd number of transitions
overall output of C $\ldots (ab)^*$
overall output of D $\ldots c(dc)^*$
the Kronecker product of C and D $\ldots 0$.

Kronecker Product for FSMs

$$C = \begin{pmatrix} \cdot & a \\ b & \cdot \end{pmatrix} \text{ and } D = \begin{pmatrix} \cdot & c & \cdot \\ \cdot & \cdot & d \\ \cdot & c & \cdot \end{pmatrix}$$

$$C \otimes D = \begin{pmatrix} \cdot & \cdot & \cdot & \cdot & ac & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & ad \\ \cdot & \cdot & \cdot & \cdot & ac & \cdot \\ \cdot & bc & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & bd & \cdot & \cdot & \cdot \\ \cdot & bc & \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

Kronecker Product for FSMs

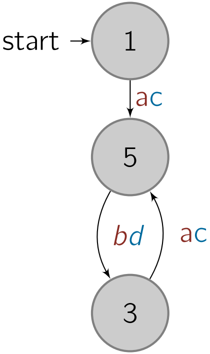


Abbildung: Graphical Representation of $C \otimes D$

the final state 2

Kronecker Product for FSMs

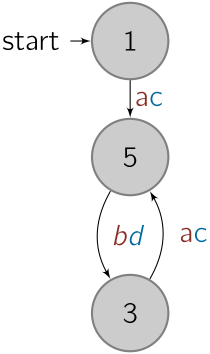


Abbildung: Graphical Representation of $C \otimes D$

the final state 2
cannot be reached from the initial state 1.

Verifying Programs

Verifying Programs

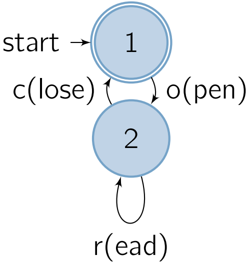


Abbildung: Graphical Representation of File Usage Scenario F

Verifying Programs

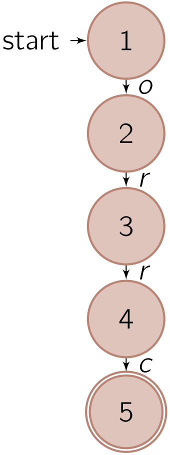


Abbildung: Graphical Representation of File Usage System A

Verifying Programs

$$A \otimes F = \begin{pmatrix} \cdot & o & \cdot & \cdot & \cdot \\ \cdot & \cdot & r & \cdot & \cdot \\ \cdot & \cdot & \cdot & r & \cdot \\ \cdot & \cdot & \cdot & \cdot & c \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} \otimes \begin{pmatrix} \cdot & o \\ c & r \end{pmatrix} = \begin{pmatrix} \cdot & \cdot & \cdot & oo & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & oc & or & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & ro & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & rc & rr & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & ro & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & rc & rr & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & co \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & cc & cr \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}.$$

Verifying Programs

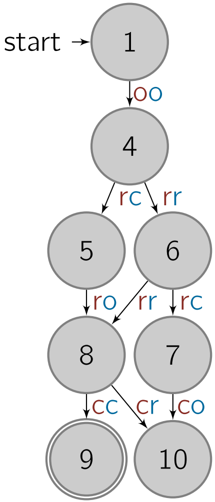


Abbildung: Graphical Representation of $A \otimes F$

Verifying Programs

pairings rc, ro, $\ldots$ do not make sense (lockstep!)

Verifying Programs

pairings rc, ro, ... do not make sense (lockstep!)
we change the definition of the Kronecker product a little bit:

Verifying Programs

pairings rc, ro, ... do not make sense (lockstep!)
we change the definition of the Kronecker product a little bit:
We assume that $rc = ro = \cdots = 0$

Verifying Programs

pairings rc, ro, … do not make sense (lockstep!)
we change the definition of the Kronecker product a little bit:
We assume that $rc = ro = \cdots = 0$
and for simplicity set $oo = o$, $rr = r$, and $cc = c$.

Verifying Programs

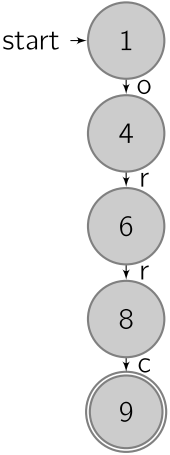


Abbildung: Graphical Representation of “new” $A \otimes F$

Verifying Programs

Figures are very similar

Verifying Programs

Figures are very similar

In fact only the node IDs differ

Verifying Programs

Figures are very similar
In fact only the node IDs differ

In graph theory, an *isomorphism* of graphs G and H is a bijection f between the node sets of G and H such that any two nodes u and v of G are adjacent in G if and only if $f(u)$ and $f(v)$ are adjacent in H. If an isomorphism exists between two graphs, then the graphs are called *isomorphic*. We write $G \simeq H$ in such a case.

Verifying Programs

Figures are very similar
In fact only the node IDs differ

In graph theory, an *isomorphism* of graphs G and H is a bijection f between the node sets of G and H such that any two nodes u and v of G are adjacent in G if and only if $f(u)$ and $f(v)$ are adjacent in H. If an isomorphism exists between two graphs, then the graphs are called *isomorphic*. We write $G \simeq H$ in such a case.

So, clearly the graphs of our example are isomorphic.

Verifying Programs

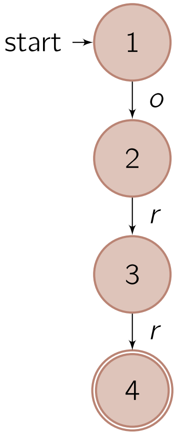


Abbildung: Graphical Representation of File Usage System B

Verifying Programs

$$B = \begin{pmatrix} \cdot & o & \cdot & \cdot \\ \cdot & \cdot & r & \cdot \\ \cdot & \cdot & \cdot & r \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix}, \; B \otimes F = \begin{pmatrix} \cdot & \cdot & \cdot & o & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & r & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & r \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

Verifying Programs

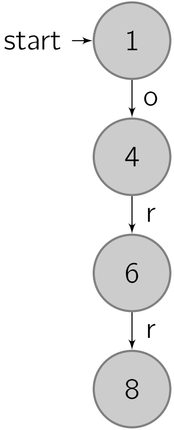


Abbildung: Graphical Representation of $B \otimes F$

Verifying Programs

Isomorphic?

Verifying Programs

Isomorphic?

Definition

An isomorphism of two control flow graphs (CFGs) G and H is a bijection f between the node sets of G and H such that any two nodes u and v of G are adjacent in G if and only if $f(u)$ and $f(v)$ are adjacent in H. In addition, let r be the root node of G. Then $f(r)$ has to be the root node of H. For all final nodes s of G, $f(s)$ have to be final nodes of H, and for all final nodes t of H, $f^{-1}(t)$ have to be final nodes in G. If an isomorphism exists between two CFGs, then the CFGs are called *isomorphic* which we denote by $G \simeq H$.

Verifying Programs

Isomorphic?

Definition

An isomorphism of two control flow graphs (CFGs) G and H is a bijection f between the node sets of G and H such that any two nodes u and v of G are adjacent in G if and only if $f(u)$ and $f(v)$ are adjacent in H. In addition, let r be the root node of G. Then $f(r)$ has to be the root node of H. For all final nodes s of G, $f(s)$ have to be final nodes of H, and for all final nodes t of H, $f^{-1}(t)$ have to be final nodes in G. If an isomorphism exists between two CFGs, then the CFGs are called *isomorphic* which we denote by $G \simeq H$.

With this definition we still have $A \otimes F \simeq A$ but $B \otimes F \not\simeq B$ because there is no final node in this case.

Verifying Programs

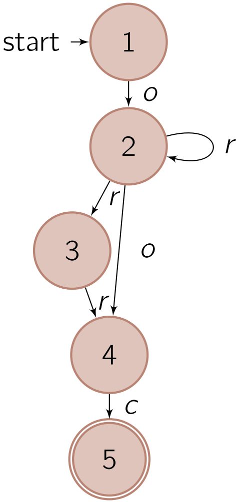


Abbildung: Graphical Representation of File Usage System C

Verifying Programs

$$C \otimes F = \begin{pmatrix} \cdot & o & \cdot & \cdot & \cdot \\ \cdot & r & r & o & \cdot \\ \cdot & \cdot & \cdot & r & \cdot \\ \cdot & \cdot & \cdot & \cdot & c \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} \otimes \begin{pmatrix} \cdot & o \\ c & r \end{pmatrix} = \begin{pmatrix} \cdot & \cdot & \cdot & o & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & o & \cdot & \cdot \\ \cdot & \cdot & \cdot & r & \cdot & r & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & r & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & c & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

Verifying Programs

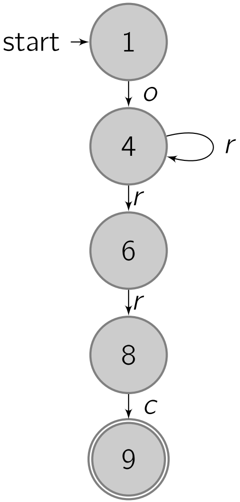


Abbildung: Graphical Representation of $C \otimes F$

Verifying Programs

Isomorphic?

Verifying Programs

Isomorphic?

General Statement

1. Assume we pick a path on program P's side, that complies to the usage scenario U. Then a corresponding path will be present in $P \otimes U$.
2. Assume we pick a path on program P's side, that does not comply completely to the usage scenario U. Then a "corresponding" path in $P \otimes U$ will end as soon as the path does not comply to U. This will result in $P \not\simeq P \otimes U$.

Verifying Programs

Isomorphic?

General Statement

1. Assume we pick a path on program P's side, that complies to the usage scenario U. Then a corresponding path will be present in $P \otimes U$.
2. Assume we pick a path on program P's side, that does not comply completely to the usage scenario U. Then a "corresponding" path in $P \otimes U$ will end as soon as the path does not comply to U. This will result in $P \not\simeq P \otimes U$.

Thus we can state:

Theorem

Given P, a control flow graph (CFG) of a program, and a usage scenario U, program P complies to U if and only if $P \simeq P \otimes U$.

The Kronecker Sum

The Kronecker Sum

- programs will use several objects (classes, modules, packages, . . .) for implementing their own task.

The Kronecker Sum

- programs will use several objects (classes, modules, packages, . . .) for implementing their own task.
- checking against each of their usage scenarios could be done one after the other.

The Kronecker Sum

- programs will use several objects (classes, modules, packages, . . .) for implementing their own task.
- checking against each of their usage scenarios could be done one after the other.
- here: check a program against two and more usage scenarios at the same time.

The Kronecker Sum

- programs will use several objects (classes, modules, packages, . . .) for implementing their own task.
- checking against each of their usage scenarios could be done one after the other.
- here: check a program against two and more usage scenarios at the same time.
- Kronecker sum.

The Kronecker Sum

Assume that $U \in M_n(\mathcal{U})$ and $W \in M_m(\mathcal{W})$ and that $\mathcal{U} \cap \mathcal{W} = \emptyset$.

The Kronecker Sum

Assume that $U \in M_n(\mathcal{U})$ and $W \in M_m(\mathcal{W})$ and that $\mathcal{U} \cap \mathcal{W} = \emptyset$.
Introduce self loops on both sides.

The Kronecker Sum

Assume that $U \in M_n(\mathcal{U})$ and $W \in M_m(\mathcal{W})$ and that $\mathcal{U} \cap \mathcal{W} = \emptyset$.
Introduce self loops on both sides.

Definition

$$U \oplus W = U \otimes \left(W + \left(\sum_{x \in \mathcal{U}} x \right) I_m \right) + \left(U + \left(\sum_{y \in \mathcal{W}} y \right) I_n \right) \otimes W =$$

$$U \otimes W + U \otimes \left(\left(\sum_{x \in \mathcal{U}} x \right) I_m \right) + U \otimes W + \left(\left(\sum_{y \in \mathcal{W}} y \right) I_n \right) \otimes W.$$

Note that $U \otimes W = Z_{n \cdot m}$ because the edge labels of U and W are disjoint. Hence we get

$$U \oplus W = U \otimes I_m + I_n \otimes W.$$

The Kronecker Sum

Theorem

With the definitions above, a program A can be checked against two usage scenarios U and W by calculating

$$A \odot (U \oplus W) = A \odot (U \otimes I_m + I_n \otimes W)\,.$$

The Kronecker Sum

We can even check a program against more than two usage scenarios at the same time.

The Kronecker Sum

We can even check a program against more than two usage scenarios at the same time. In order to be able to write this in a concise way, we introduce

$$\bigoplus_{x \in \mathcal{X}} x$$

similar to the sigma notation for standard sums.

The Kronecker Sum

We can even check a program against more than two usage scenarios at the same time. In order to be able to write this in a concise way, we introduce

$$\bigoplus_{x \in \mathcal{X}} x$$

similar to the sigma notation for standard sums.

Theorem

With the definitions above, a program A can be checked against usage scenarios U_i where $i = 1, \ldots, k$ by calculating

$$A \odot \left(\bigoplus_{i=1}^{k} U_i \right).$$

The Kronecker Sum

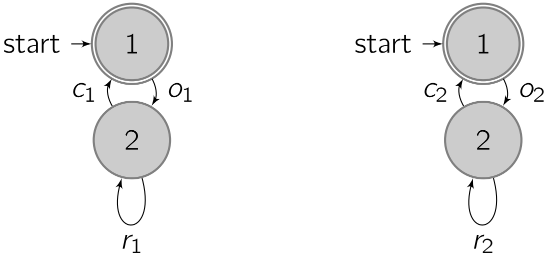


Abbildung: Graphical Representation of File Usage Scenarios U and W

The Kronecker Sum

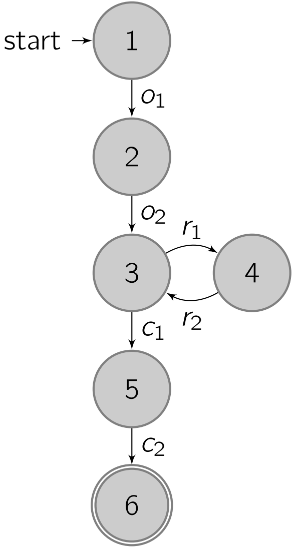


Abbildung: CFG of Program A

The Kronecker Sum

$$U = \begin{pmatrix} \cdot & o_1 \\ c_1 & r_1 \end{pmatrix},$$
$$W = \begin{pmatrix} \cdot & o_2 \\ c_2 & r_2 \end{pmatrix},$$

and

$$A = \begin{pmatrix} \cdot & o_1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & o_2 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & r_1 & c_1 & \cdot \\ \cdot & \cdot & r_2 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & c_2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}.$$

The Kronecker Sum

$$
\begin{aligned}
U \oplus W =& U \otimes I_2 + I_2 \otimes W = \\
& \begin{pmatrix} \cdot & o_1 \\ c_1 & r_1 \end{pmatrix} \otimes \begin{pmatrix} 1 & \cdot \\ \cdot & 1 \end{pmatrix} + \begin{pmatrix} 1 & \cdot \\ \cdot & 1 \end{pmatrix} \otimes \begin{pmatrix} \cdot & o_2 \\ c_2 & r_2 \end{pmatrix} = \\
& \begin{pmatrix} \cdot & \cdot & o_1 & \cdot \\ \cdot & \cdot & \cdot & o_1 \\ c_1 & \cdot & r_1 & \cdot \\ \cdot & c_1 & \cdot & r_1 \end{pmatrix} + \begin{pmatrix} \cdot & o_2 & \cdot & \cdot \\ c_2 & r_2 & \cdot & \cdot \\ \cdot & \cdot & \cdot & o_2 \\ \cdot & \cdot & c_2 & r_2 \end{pmatrix} = \\
& \begin{pmatrix} \cdot & o_2 & o_1 & \cdot \\ c_2 & r_2 & \cdot & o_1 \\ c_1 & \cdot & r_1 & o_2 \\ \cdot & c_1 & c_2 & r_1 + r_2 \end{pmatrix} .
\end{aligned}
$$

The Kronecker Sum

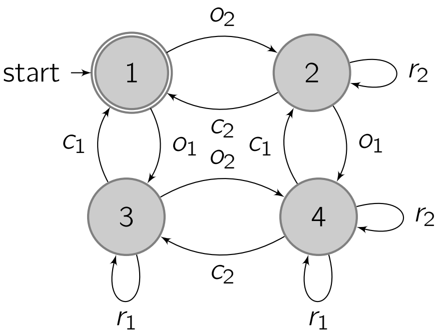


Abbildung: Graphical Representation of $U \oplus W$

The Kronecker Sum

$A \odot (U \oplus W) =$

$$\begin{pmatrix}
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & o_1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & o_1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & o_2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & o_2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & r_1 & \cdot & c_1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & r_1 & \cdot & c_1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & r_2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & r_2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & c_2 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & c_2 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot
\end{pmatrix}.$$

The Kronecker Sum

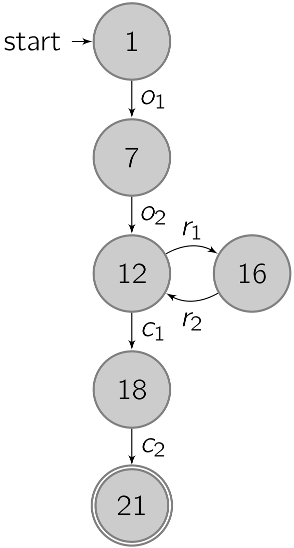


Abbildung: Graphical Representation of $A \odot (U \oplus W)$

The Kronecker Sum

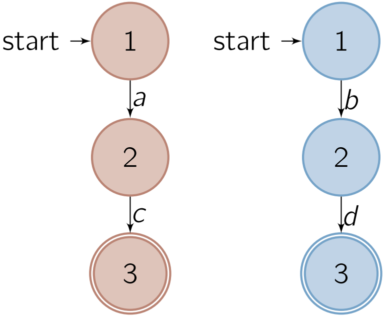


Abbildung: Our introductory example from a previous section

Kronecker Sum

Matrices

$$A = \begin{pmatrix} \cdot & a & \cdot \\ \cdot & \cdot & c \\ \cdot & \cdot & \cdot \end{pmatrix}$$

and

$$B = \begin{pmatrix} \cdot & b & \cdot \\ \cdot & \cdot & d \\ \cdot & \cdot & \cdot \end{pmatrix}$$

Kronecker Sum

Kronecker Product 1

$$A \otimes I_3 = \begin{pmatrix} \cdot & \cdot & \cdot & a & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & a & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & a & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & c & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & c & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & c \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}.$$

Kronecker Sum

Kronecker Product 2

$$I_3 \otimes B = \begin{pmatrix} \cdot & b & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & d & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & b & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & d & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & b & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & d \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix} .$$

Kronecker Sum

Kronecker Sum

$$A \otimes I_3 + I_3 \otimes B = \begin{pmatrix} \cdot & b & \cdot & a & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & d & \cdot & a & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & a & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & b & \cdot & c & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & d & \cdot & c & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & c \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & b & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & d \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}.$$

The Kronecker Sum

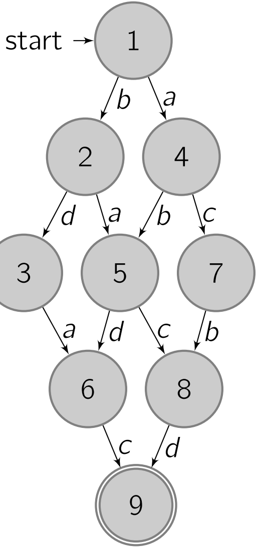


Abbildung: Resulting Interleavings Graph

The Kronecker Sum

For more on Kronecker Algebra and its many applications see `https://kronalg.blieberger.at`.