Towards Interactive Verification of Programmable Logic Controllers using Modal Kleene Algebra and KIV

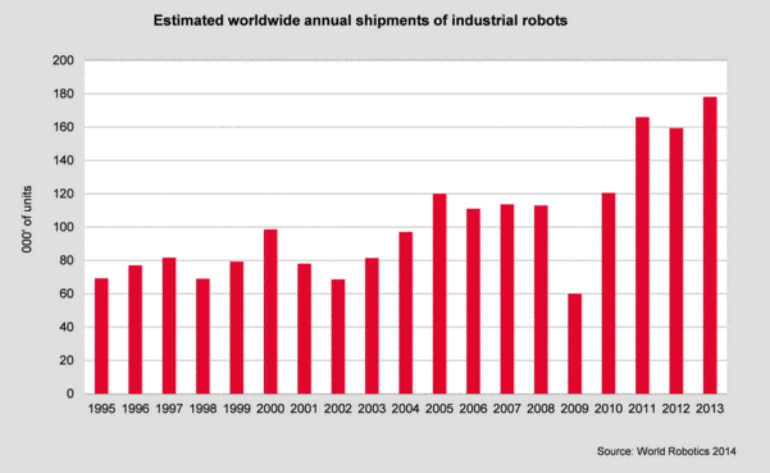
Roland Glück, Florian Benedikt Krebs

ST-BT

Braga, September 28, 2015

Outline

1. Introduction
2. PLC Crash Course
3. Modal Kleene Algebra and Linear Temporal Logic
4. Function Block Diagrams in Modal Kleene Algebra
5. Case Study: Mutual Exclusion
6. Conclusion and Outlook


Estimated worldwide annual shipments of industrial robots
000' of units
200
180
160
140
120
100
80
60
40
20
0
1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013
Source: World Robotics 2014

robots are:

robots are:

- cost saving

robots are:

- cost saving
- reliable

robots are:

- cost saving
- reliable
- strong

robots are:

- cost saving
- reliable
- strong
- very strong

robots are:

- cost saving
- reliable
- strong
- very strong
- insensitive

robots are:

- cost saving
- reliable
- strong
- very strong
- insensitive
- dangerous

$\Rightarrow$ careful control is indispensable

PLC - Purpose and Function

- Programmable Logic Controllers (PLCs) used for controlling various plants
- robots, pumps, valves, mechanical and automated devices, ...
- PLC works in cyclic way (1 - 150 ms):
  - reads input channels (sensors, switches, internal variables)
  - computes new values
  - writes new values to associated output channels/registers (actuators, internal variables)

Data Types and Safety

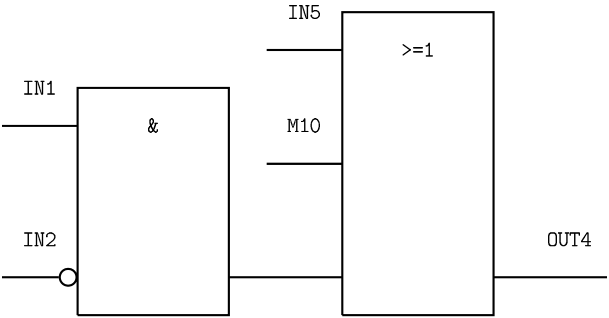
- possible data types: `bool`, `int`, `float`, `date`, ...
- with usual operations (numerical, comparision, ...)
- special part for safety critical operations with reduced instruction set
- from now on only Boolean data and operations

Programming Languages

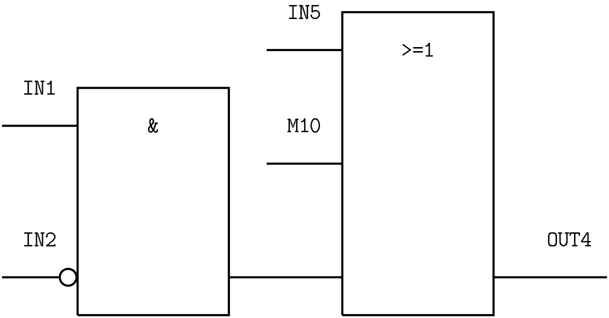
Programming done via:

- Instruction List (IL): assembly-like
- Ladder Diagram (LD): similar to circuit diagrams
- Sequential Function Chart (SFC): inspired by state diagrams
- Structured Text (ST): resembles C syntax
- Function Block Diagram (FBD): see next

AND, OR and Negation in FBD



AND, OR and Negation in FBD



$$\mathtt{OUT4} \equiv (\mathtt{IN1} \wedge \neg \mathtt{IN2}) \vee \mathtt{IN5} \vee \mathtt{M10}$$

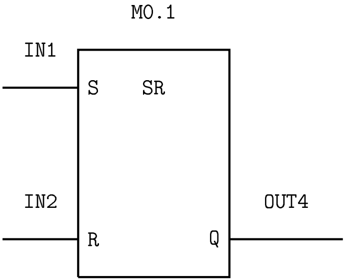
Flip-Flops (Purpose and Function)

- Flip-Flops show dynamic behavior
- two inputs and one output
- `TRUE`-signal on set input sets output persistently to `TRUE`
- `TRUE`-signal on reset input resets output persistently to `FALSE`
- (until next signal on set/reset input)
- set/reset dominant depending on winner at set/reset conflict
- storing/clearing depending on input signals

Flip-Flops (Truth Table)

S_n	R_n	Q_{n+1}
TRUE	FALSE	TRUE
FALSE	TRUE	FALSE
FALSE	FALSE	Q_n
TRUE	TRUE	TRUE (set dominant)
TRUE	TRUE	FALSE (reset dominant)

Flip-Flops (FBD)



Kleene Algebra

Definition

A *Kleene algebra* is a structure $(M, +, 0, \cdot, 1, ^*)$ where $(M, +, 0, \cdot, 1)$ is an idempotent semiring and $^* : M \to M$ has the following properties:

$$1 + xx^* \leq x^* \qquad x + yz \leq z \Rightarrow y^*x \leq z$$
$$1 + x^*x \leq x^* \qquad x + yz \leq y \Rightarrow xz^* \leq y$$

- $+$ models choice, $\cdot$ composition, * iteration
- natural order defined by $x \leq y \Leftrightarrow_{df} x + y = y$
- examples: formal languages, relations, ...

Tests

given an idempotent semiring $S = (M, +, 0, \cdot, 1)$ subsets of M can be modeled by tests:

Definition

Given an idempotent semiring $S = (M, +, 0, \cdot, 1)$ an element $p \in M$ is called a *test* if an element $\neg p$ (the *complement* of p) exists with the properties $p + \neg p = 1$ and $p \cdot \neg p = 0 = \neg p \cdot p$.

- set of tests denoted by $\mathbf{test}(S)$
- in relational context: subsets of identity

Boxes and Diamonds

(pre)image or (pre I post)condition modeled by diamond/box operators:

Definition

A *modal semiring* is a structure $S = (M, +, 0, \cdot, 1, |\cdot\rangle, \langle\cdot|)$ where $S' = (M, +, 0, \cdot, 1)$ is an idempotent semiring and $|\cdot\rangle$ and $\langle\cdot|$ are functions of the type $M \to (\textbf{test}(S') \to \textbf{test}(S'))$ with the properties $|x\rangle p \leq q \Leftrightarrow \neg qxp \leq 0 \Leftrightarrow \langle x|p \leq \neg q$, $|xy\rangle p = |x\rangle|y\rangle p$ and $\langle xy|p = \langle y|\langle x|p$ for all $x \in M$ and $p, q \in S'$.

- $|a\rangle p$: transition into p is possible
- $|a]p =_{df} \neg|a\rangle\neg p$: transition into p is inevitable

Modal Kleene Algebra

putting all together:

Definition

A *modal Kleene algebra* (*MKA* for short) is a structure $(M, +, 0, \cdot, 1, |\cdot\rangle, \langle\cdot|, {}^*)$ where $(M, +, 0, \cdot, 1, |\cdot\rangle, \langle\cdot|)$ is a modal semiring and $(M, +, 0, \cdot, 1, {}^*)$ is a Kleene algebra.

Modal Kleene Algebra and Linear Temporal Logic

work by Möller, Höfner and Struth (2006):

- model transition system by a general MKA element a
- transforming sets of traces into sets of successors
- left total function modeled by $|a\rangle p = |a]p$ for all tests p
- formulae in linear temporal logic (LTL) correspond to expressions in MKA
- LTL formula is valid iff corresponding MKA expression evaluates to 1

Explicit Correspondence

$$
\begin{array}{lcll}
[\![\bot]\!] & = & 0 & \\
[\![\neg\psi]\!] & = & \neg[\![\psi]\!] & \\
[\![\psi_1 \wedge \psi_2]\!] & = & [\![\psi_1]\!]\cdot[\![\psi_2]\!] & \\
[\![\psi_1 \vee \psi_2]\!] & = & [\![\psi_1]\!]+[\![\psi_2]\!] & \\
[\![\psi_1 \rightarrow \psi_2]\!] & = & [\![\psi_1]\!]\rightarrow[\![\psi_2]\!] & (p \rightarrow q =_{df} \neg p + q) \\
[\![\Box\,\psi]\!] & = & [\![|a^*]\psi]\!] & \\
[\![\Diamond\,\psi]\!] & = & [\![|a^*\rangle\psi]\!] & \\
[\![\circ\psi]\!] & = & [\![|a\rangle\psi]\!] & \\
[\![\psi_1 \,\mathrm{U}\, \psi_2]\!] & = & |([\![\psi_1]\!]\cdot a)^*\rangle[\![\psi_2]\!] &
\end{array}
$$

Variables and Overall Behavior

FBDs in MKA:

Variables and Overall Behavior

FBDs in MKA:

- inputs/outputs/internal variables correspond to tests
- for every signal/variable `p` introduce two tests `p_0` and `p_1`
- indicating a value of `FALSE` and `TRUE`, resp.
- clearly $\neg$`p_0` $=$ `p_1` and $\neg$`p_1` $=$ `p_0`
- characterize behavior of elementary gates (`OR`, `AND`, Flip-Flops, ...)
- elementary gates do not change noninvolved signals/variables
- remember left total functionality
- write overall behavior *a* as product of elementary gates

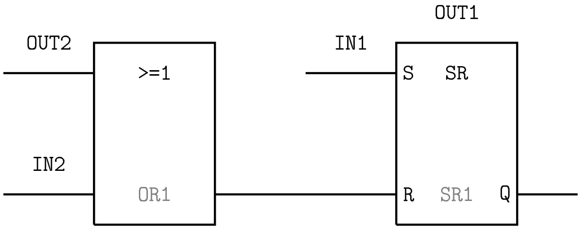
Elementary Gates

- $\mathtt{AND}$-gate $\mathtt{ANDk}$ with inputs $\mathtt{in1,\ in2}\ \ldots, \mathtt{inn}$:
  - $\mathtt{in1\_1} \cdot \mathtt{in2\_1} \cdot \cdots \cdot \mathtt{inn\_1} \leq |\mathtt{andk}\rangle \mathtt{andk\_1}$
  - $\mathtt{in1\_0} + \mathtt{in2\_0} + \cdots + \mathtt{inn\_0} \leq |\mathtt{andk}\rangle \mathtt{andk\_0}$.
- $\mathtt{OR}$-gate $\mathtt{ORk}$ with inputs $\mathtt{in1,\ in2}\ \ldots, \mathtt{inn}$:
  - $\mathtt{in1\_1} + \mathtt{in2\_1} + \cdots + \mathtt{inn\_1} \leq |\mathtt{ork}\rangle \mathtt{ork\_1}$
  - $\mathtt{in1\_0} \cdot \mathtt{in2\_0} \cdot \cdots \cdot \mathtt{inn\_0} \leq |\mathtt{ork}\rangle \mathtt{ork\_0}$.
- negation of $\mathtt{sk}$: switch $\mathtt{sk\_1}$ and $\mathtt{sk\_0}$

Flip-Flops

- set dominant flip-flop RSk with set input s, reset input r, output q and internal marker m :
  - $\mathtt{s\_1} + \mathtt{m\_1} \cdot \mathtt{r\_0} \leq |\mathtt{rsk}\rangle \mathtt{q\_1}$
  - $\mathtt{s\_1} + \mathtt{m\_1} \cdot \mathtt{r\_0} \leq |\mathtt{rsk}\rangle \mathtt{m\_1}$
  - $\mathtt{s\_0} \cdot \mathtt{r\_1} + \mathtt{m\_0} \cdot \mathtt{s\_0} \leq |\mathtt{rsk}\rangle \mathtt{q\_0}$
  - $\mathtt{s\_0} \cdot \mathtt{r\_1} + \mathtt{m\_0} \cdot \mathtt{s\_0} \leq |\mathtt{rsk}\rangle \mathtt{m\_0}$

Example Construction (not Complete!)



$$\mathtt{out2\_1} + \mathtt{in2\_1} \leq |\mathtt{or1}\rangle \mathtt{or1\_1}$$

$$\mathtt{out2\_0} \cdot \mathtt{in2\_0} \leq |\mathtt{or1}\rangle \mathtt{or1\_0}$$

$$\mathtt{in1\_0} \leq |\mathtt{or1}\rangle \mathtt{in1\_0}$$

$$\mathtt{in1\_1} \leq |\mathtt{or1}\rangle \mathtt{in1\_1}$$

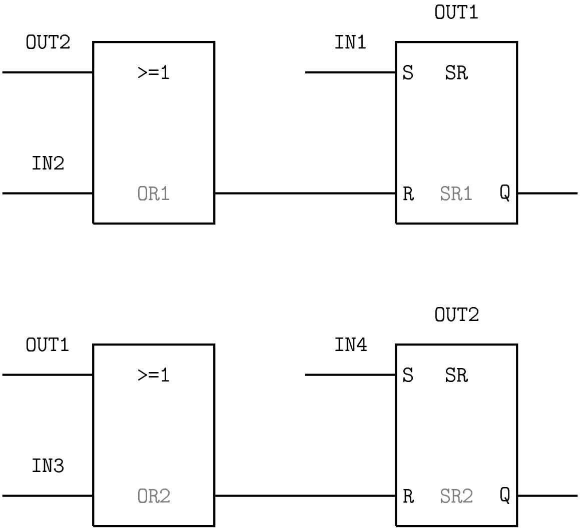
$$|\mathtt{or1}\rangle \mathtt{p} = |\mathtt{or1}]\mathtt{p}$$

$$\mathtt{or1\_1} + \mathtt{out1\_0} \cdot \mathtt{in1\_0} \leq |\mathtt{sr1}\rangle \mathtt{out1\_0}$$

$$\mathtt{in1\_1} \cdot \mathtt{or1\_0} + \mathtt{out1\_1} \cdot \mathtt{or1\_0} \leq |\mathtt{sr1}\rangle \mathtt{out1\_1}$$

$$\mathtt{cycle} = \mathtt{or1} \cdot \mathtt{sr1}$$

Mutual Exclusion



Behavior and Desired Properties

- behavior given by $\mathtt{cycle} = \mathtt{or1} \cdot \mathtt{sr1} \cdot \mathtt{or2} \cdot \mathtt{sr2}$

Behavior and Desired Properties

- behavior given by $\mathtt{cycle} = \mathtt{or1} \cdot \mathtt{sr1} \cdot \mathtt{or2} \cdot \mathtt{sr2}$
- desired properties in LTL:
  - $\mathtt{out1\_0} \cdot \mathtt{out2\_0} \rightarrow \Box(\mathtt{out1\_1} \rightarrow \mathtt{out2\_0})$
  - $\mathtt{out1\_0} \cdot \mathtt{out2\_0} \rightarrow \Box(\mathtt{out2\_1} \rightarrow \mathtt{out1\_0})$

Behavior and Desired Properties

- behavior given by $\mathtt{cycle} = \mathtt{or1} \cdot \mathtt{sr1} \cdot \mathtt{or2} \cdot \mathtt{sr2}$
- desired properties in LTL:
  - $\mathtt{out1\_0} \cdot \mathtt{out2\_0} \to \Box(\mathtt{out1\_1} \to \mathtt{out2\_0})$
  - $\mathtt{out1\_0} \cdot \mathtt{out2\_0} \to \Box(\mathtt{out2\_1} \to \mathtt{out1\_0})$
- in MKA (recall $p \to q =_{df} \neg p + q$):
  - $\mathtt{out1\_0} \cdot \mathtt{out2\_0} \to |\mathtt{cycle}^*](\mathtt{out1\_1} \to \mathtt{out2\_0}) = 1$
  - $\mathtt{out1\_0} \cdot \mathtt{out2\_0} \to |\mathtt{cycle}^*](\mathtt{out2\_1} \to \mathtt{out1\_0}) = 1$

Proof Sketch

to show: $\texttt{out1\_0} \cdot \texttt{out2\_0} \to |\texttt{cycle}^*](\texttt{out1\_1} \to \texttt{out2\_0}) = 1$

Proof Sketch

to show: $\mathtt{out1\_0} \cdot \mathtt{out2\_0} \to |\mathtt{cycle}^*](\mathtt{out1\_1} \to \mathtt{out2\_0}) = 1$

proof sketch:

- first: $\mathtt{out1\_0} \cdot \mathtt{out2\_0} + \mathtt{out1\_0} \cdot \mathtt{out2\_1} + \mathtt{out1\_1} \cdot \mathtt{out2\_0}$ is an invariant of `cycle`

Proof Sketch

to show: $\texttt{out1\_0} \cdot \texttt{out2\_0} \to |\texttt{cycle}^*](\texttt{out1\_1} \to \texttt{out2\_0}) = 1$

proof sketch:

- first: $\texttt{out1\_0} \cdot \texttt{out2\_0} + \texttt{out1\_0} \cdot \texttt{out2\_1} + \texttt{out1\_1} \cdot \texttt{out2\_0}$ is an invariant of `cycle`
- MKA: $\texttt{out1\_0} \cdot \texttt{out2\_0} + \texttt{out1\_0} \cdot \texttt{out2\_1} + \texttt{out1\_1} \cdot \texttt{out2\_0}$ is an invariant of $\texttt{cycle}^*$

Proof Sketch

to show: $\mathtt{out1\_0} \cdot \mathtt{out2\_0} \rightarrow |\mathtt{cycle}^*](\mathtt{out1\_1} \rightarrow \mathtt{out2\_0}) = 1$

proof sketch:

- first: $\mathtt{out1\_0} \cdot \mathtt{out2\_0} + \mathtt{out1\_0} \cdot \mathtt{out2\_1} + \mathtt{out1\_1} \cdot \mathtt{out2\_0}$ is an invariant of `cycle`
- MKA: $\mathtt{out1\_0} \cdot \mathtt{out2\_0} + \mathtt{out1\_0} \cdot \mathtt{out2\_1} + \mathtt{out1\_1} \cdot \mathtt{out2\_0}$ is an invariant of $\mathtt{cycle}^*$
- MKA: $p \leq q \wedge qx\neg q = 0 \wedge q \leq r \Rightarrow p \rightarrow |x]r = 1$

Proof Sketch

to show: $\mathtt{out1\_0} \cdot \mathtt{out2\_0} \to |\mathtt{cycle}^*](\mathtt{out1\_1} \to \mathtt{out2\_0}) = 1$

proof sketch:

- first: $\mathtt{out1\_0} \cdot \mathtt{out2\_0} + \mathtt{out1\_0} \cdot \mathtt{out2\_1} + \mathtt{out1\_1} \cdot \mathtt{out2\_0}$ is an invariant of `cycle`
- MKA: $\mathtt{out1\_0} \cdot \mathtt{out2\_0} + \mathtt{out1\_0} \cdot \mathtt{out2\_1} + \mathtt{out1\_1} \cdot \mathtt{out2\_0}$ is an invariant of $\mathtt{cycle}^*$
- MKA: $p \leq q \wedge qx\neg q = 0 \wedge q \leq r \Rightarrow p \to |x]r = 1$
- finish:
  - $\mathtt{out1\_0} \cdot \mathtt{out2\_0} \leq \mathtt{out1\_0} \cdot \mathtt{out2\_0} + \mathtt{out1\_0} \cdot \mathtt{out2\_1} + \mathtt{out1\_1} \cdot \mathtt{out2\_0}$
  - $\mathtt{out1\_0} \cdot \mathtt{out2\_0} + \mathtt{out1\_0} \cdot \mathtt{out2\_1} + \mathtt{out1\_1} \cdot \mathtt{out2\_0} \leq \mathtt{out1\_1} \to \mathtt{out2\_0}$

Proof Sketch

to show: $\texttt{out1\_0} \cdot \texttt{out2\_0} \to |\texttt{cycle}^*](\texttt{out1\_1} \to \texttt{out2\_0}) = 1$

proof sketch:

- first: $\texttt{out1\_0} \cdot \texttt{out2\_0} + \texttt{out1\_0} \cdot \texttt{out2\_1} + \texttt{out1\_1} \cdot \texttt{out2\_0}$ is an invariant of $\texttt{cycle}$
- MKA: $\texttt{out1\_0} \cdot \texttt{out2\_0} + \texttt{out1\_0} \cdot \texttt{out2\_1} + \texttt{out1\_1} \cdot \texttt{out2\_0}$ is an invariant of $\texttt{cycle}^*$
- MKA: $p \leq q \land qx\neg q = 0 \land q \leq r \Rightarrow p \to |x]r = 1$
- finish:
  - $\texttt{out1\_0} \cdot \texttt{out2\_0} \leq \texttt{out1\_0} \cdot \texttt{out2\_0} + \texttt{out1\_0} \cdot \texttt{out2\_1} + \texttt{out1\_1} \cdot \texttt{out2\_0}$
  - $\texttt{out1\_0} \cdot \texttt{out2\_0} + \texttt{out1\_0} \cdot \texttt{out2\_1} + \texttt{out1\_1} \cdot \texttt{out2\_0} \leq \texttt{out1\_1} \to \texttt{out2\_0}$
- proof done interactively in KIV

Conclusion

We saw:

Conclusion

We saw:

- Programmable Logic Controllers

Conclusion

We saw:

- Programmable Logic Controllers
- Modal Kleene Algebra

Conclusion

We saw:

- Programmable Logic Controllers
- Modal Kleene Algebra
- Linear Temporal Logic

Conclusion

We saw:

- Programmable Logic Controllers
- Modal Kleene Algebra
- Linear Temporal Logic
- interactive proving with KIV

Conclusion

We saw:

- Programmable Logic Controllers
- Modal Kleene Algebra
- Linear Temporal Logic
- interactive proving with KIV
- and all working together

Outlook

We plan:

Outlook

We plan:

- verification of real safety systems

Outlook

We plan:

- verification of real safety systems
- typical features:
  - 32 - 64 signals from sensors
  - plus up to 16 signals from safety doors
  - 50 - 100 elementary gates

Outlook

We plan:

- verification of real safety systems
- typical features:
  - 32 - 64 signals from sensors
  - plus up to 16 signals from safety doors
  - 50 - 100 elementary gates
- characterization of other gates in MKA

Outlook

We plan:

- verification of real safety systems
- typical features:
  - 32 - 64 signals from sensors
  - plus up to 16 signals from safety doors
  - 50 - 100 elementary gates
- characterization of other gates in MKA
- embracing numerical operations

Outlook

We plan:

- verification of real safety systems
- typical features:
  - 32 - 64 signals from sensors
  - plus up to 16 signals from safety doors
  - 50 - 100 elementary gates
- characterization of other gates in MKA
- embracing numerical operations
- timer

Outlook

We plan:

- verification of real safety systems
- typical features:
  - 32 - 64 signals from sensors
  - plus up to 16 signals from safety doors
  - 50 - 100 elementary gates
- characterization of other gates in MKA
- embracing numerical operations
- timer
- automated construction of input files

Obrigado pela atenção!

Obrigado pela atenção!

Perguntas?