A Survey of Reactive Synthesis

Nikolaj S. Bjørner
Dept. of Computer Science
Stanford University
Advisor: Zohar Manna

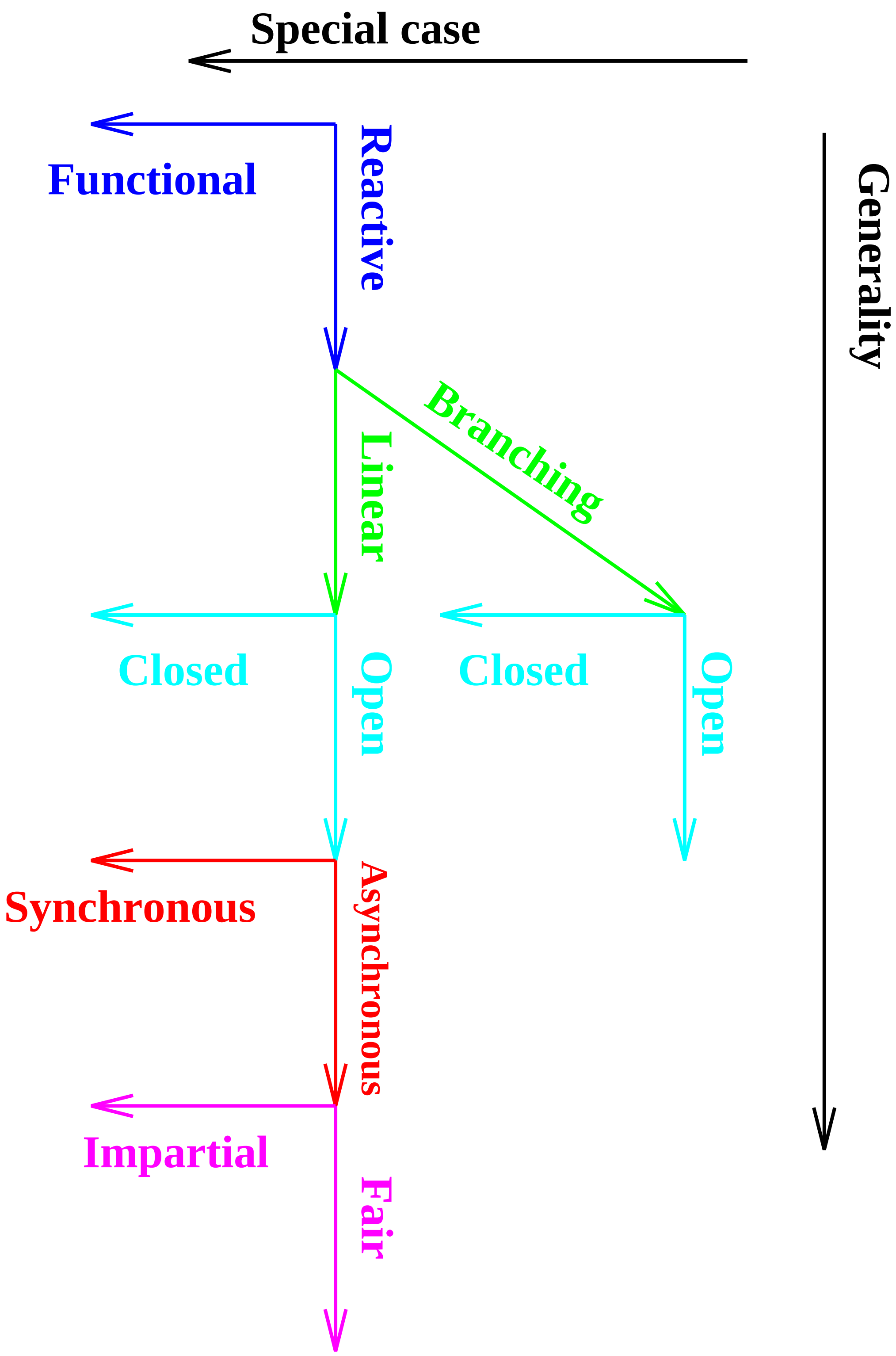
Problem:

Derive reactive module
from temporal specification

Solutions:

Automata theory
Model building
Proof theory
Constraint satisfaction

Road Map



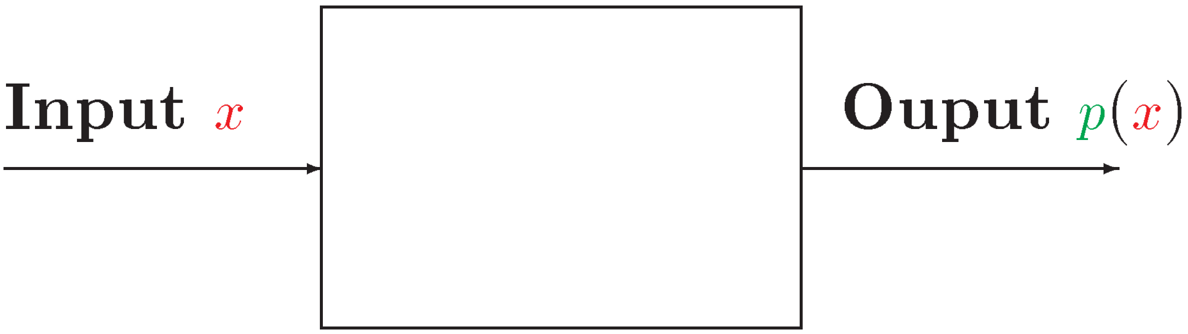
Special case
Generality
Functional
Reactive
Branching
Linear
Closed
Open
Closed
Open
Synchronous
Asynchronous
Impartial
Fair

Functional Synthesis

First-order specification $\forall x\ \exists y \cdot \varphi(x, y)$

Find program $p(x)$ such that

$$\forall x \cdot \varphi(x, p(x))$$



- Extraction of programs from proofs
- Program transformations

Works on Functional Synthesis

[Curry-Howard 69,80]	Formulas as types
[Burstall-Darlington 77]	Transformations
[Manna-Waldinger 80]	Tableau
[Mints-Tyugu 82]	Priz
[Coquand-Huet 85]	Coq
[Constable 86]	Nuprl
[Hayashi-Nakano 88]	PX
[Smith 90]	KIDS
[Abramsky 90]	Linear Logic
[Murthy 92]	Continuations
[Lowry 95]	Amphion

Functional Synthesis Example

Prove

$$\forall x,\ \epsilon > 0\ \exists y \cdot (y^2 \leq x < (y+\epsilon)^2)$$

Extract

$$
\begin{aligned}
&\mathbf{sqrt}(x,\epsilon) = \\
&\qquad \text{if } \epsilon^2 > x \ \text{ then } 0 \ \text{ else} \\
&\qquad \text{let } \ z = \mathbf{sqrt}(x, 2\cdot\epsilon) \text{ in} \\
&\qquad\qquad \text{if } \ (z+\epsilon)^2 \leq x \\
&\qquad\qquad \text{then } \ z+\epsilon \\
&\qquad\qquad \text{else } \ z
\end{aligned}
$$

From a constructive proof

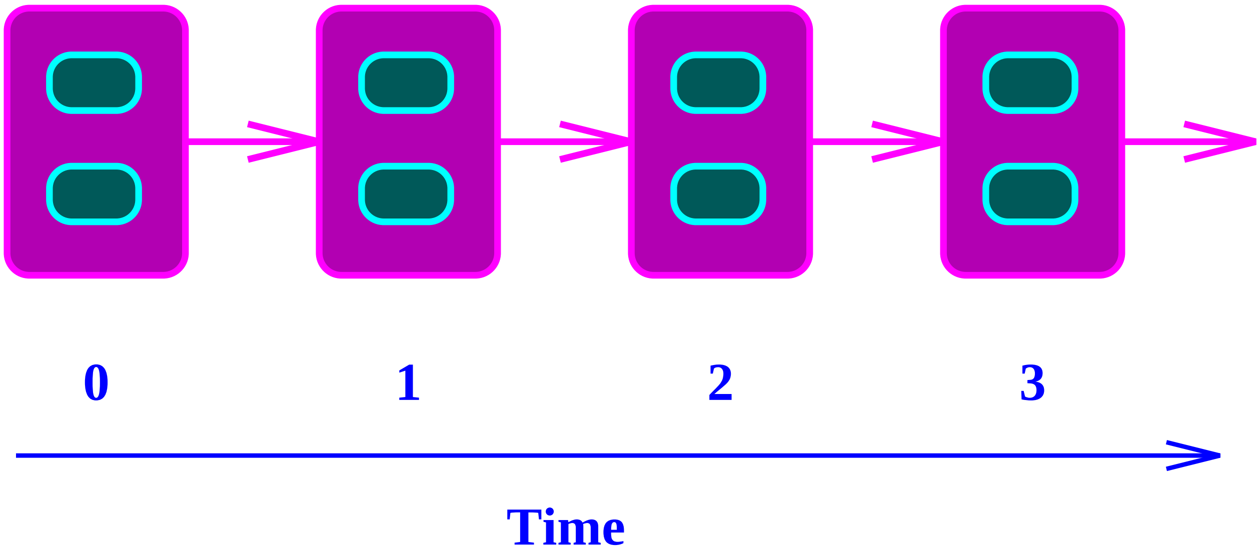
Induction = Recursion

Resolution = Case distinction

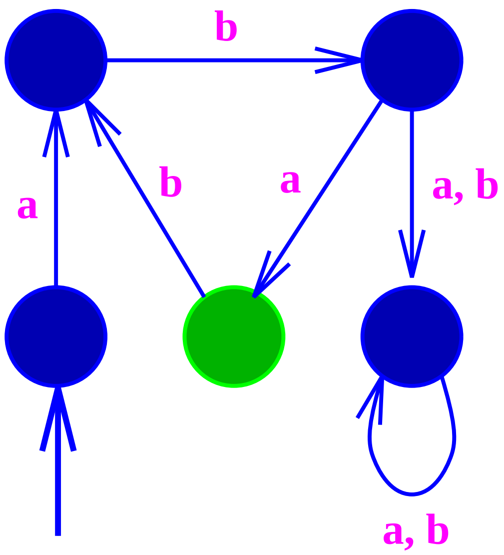
Cut = Let-binding

Reactive Synthesis

Characterized by ongoing behaviour

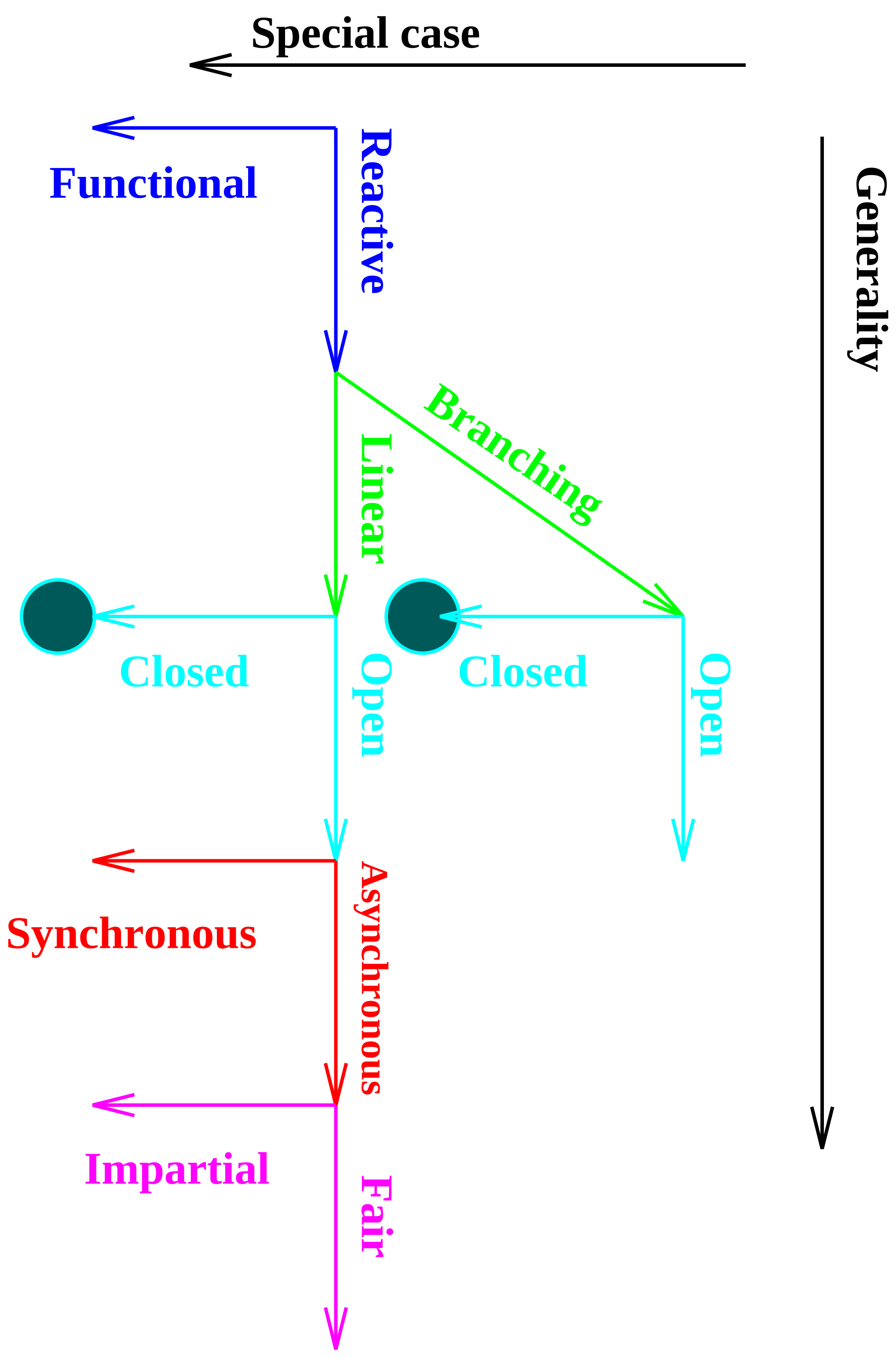


Specification φ is temporal



$a;$
loop forever do
$\left[b;\ \ a;\ \ b\ \right]$

Road Map



Special case
Functional
Reactive
Generality
Branching
Linear
Closed
Open
Closed
Open
Synchronous
Asynchronous
Impartial
Fair

Closed System Synthesis

Satisfiability problem

$$\exists y \cdot \varphi(y)$$

LTL solution [Manna-Wolper 81]

- Build tableau (PSPACE)
- Check non-emptiness (LINEAR)
- Extract CSP program from model

CTL solution [Clarke-Emerson 81]

- Build branching tableau (EXPTIME)
- Check non-emptiness (QUADRATIC)
- Extract test-and-set from model

Closed System Synthesis

ω-regular solution
[Anuchitanukul 95 (thesis)]

- Build Büchi automaton $\mathcal{B}$ (PSPACE)
- Extract fair transition system $\mathcal{T}$
  Nondeterministic (LINEAR)
  Each transition deterministic (EXPONENTIAL)

Features:

- Maximal synthesis

$$L(\mathcal{T}) = L(\mathcal{B})$$

- Minimal degree of parallelism
  - static
    FTS $\Rightarrow$ $|\mathcal{T}| = 2$
  - dynamic
    DSTS $\Rightarrow$ optimization algorithm

Hierarchy of Fair Transition Systems

FTS	Fair transition system
2FTS	FTS with 2 transitions
Buchi	ω-regular
STS	Compassionate
WTS	Just
DTS	Deterministic
DSTS	Deterministic compassionate
D-Buchi	
DWTS	Deterministic just

Closed Multiprocess Synthesis

[Attie-Emerson 89]

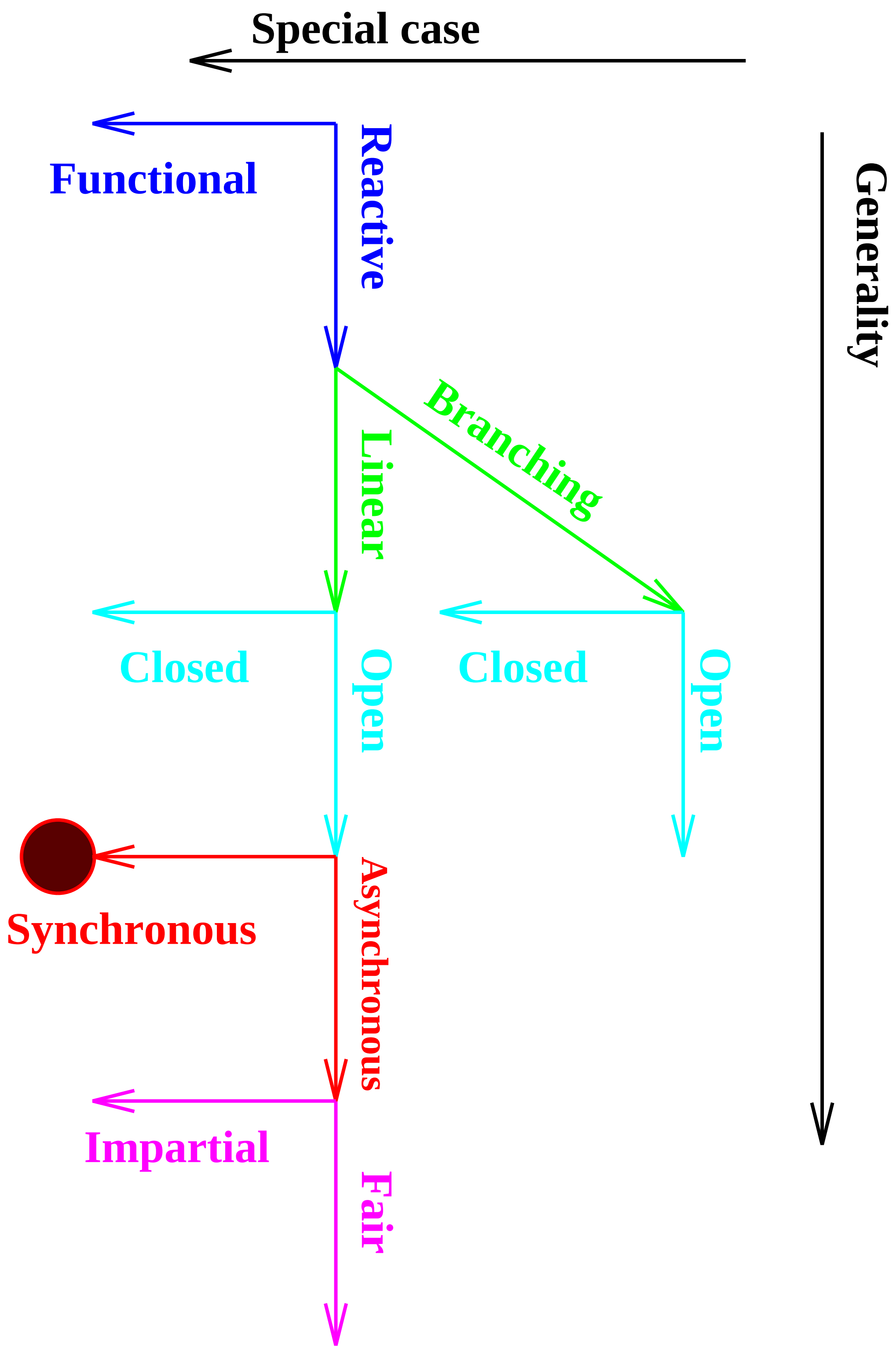
Problem: Combinatorial explosion encountered for 3-4 processes

Solution: Use multiprocess CTL with extra process quantifiers:

$$\bigwedge_{i,j} (R(i,j) \rightarrow \varphi) \qquad\qquad \bigvee_{i,j} (R(i,j) \wedge \varphi)$$

Benefit: Uniform synchronization skeletons are derived for any number of processes

Road Map



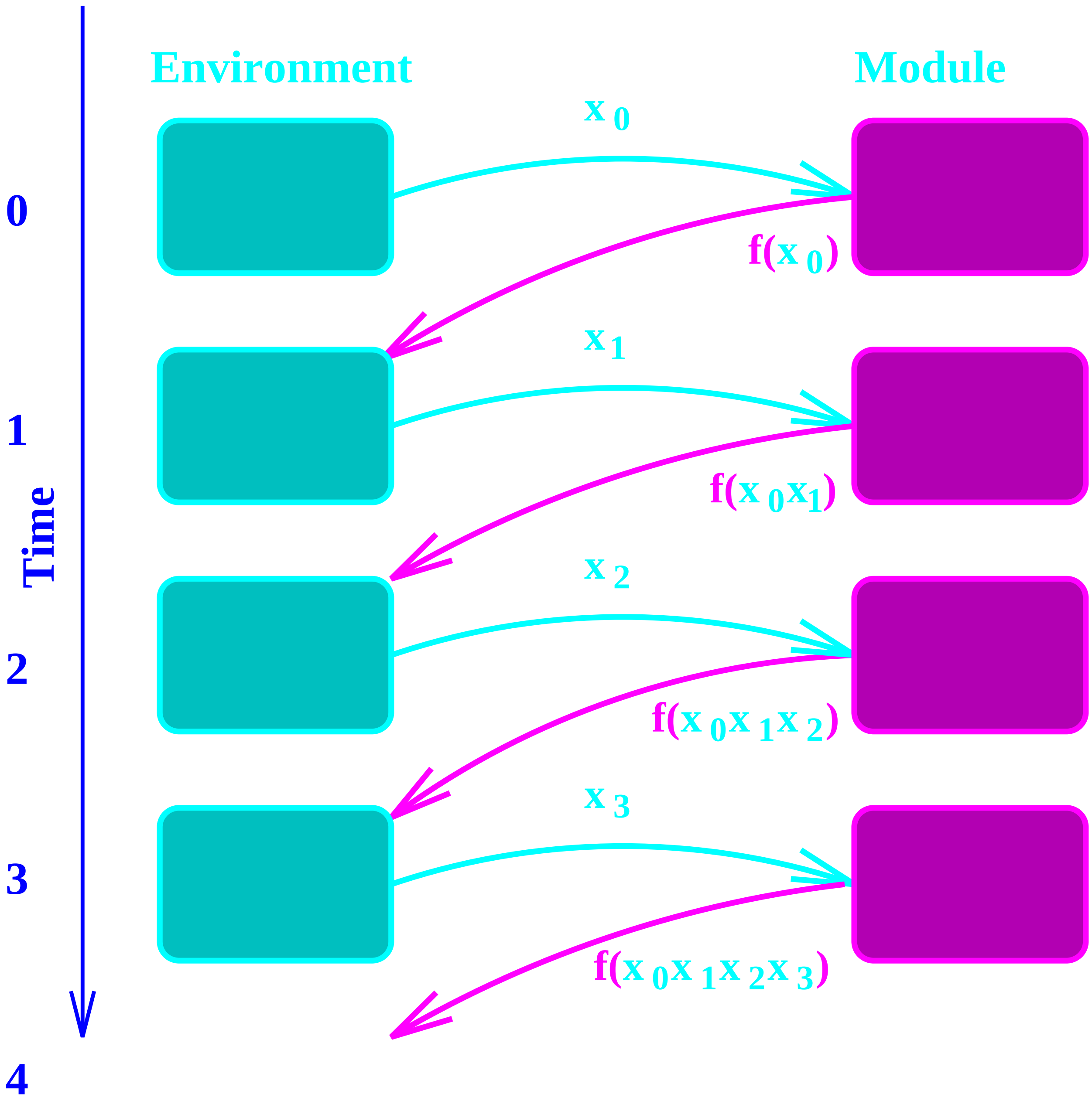
Special case
Functional
Reactive
Generality
Branching
Linear
Closed
Open
Closed
Open
Synchronous
Asynchronous
Impartial
Fair

Open Synchronous

Temporal specification $\forall x \; \exists y \cdot \varphi(x, y)$.

Find program $f(x)$ such that

$$\forall x \cdot \varphi(x, f(x))$$



Church's Problem [Church 62]

Summary of early digital synthesis and verification

Specification $\mathcal{C}(X : I^\omega, Y : J^\omega)$ in "restricted recursive arithmetic"

Find operator $f : I^\omega \to J^\omega$
such that

$$\forall X \cdot \mathcal{C}(X, f(X))$$

is valid

Requirements

(I) f may not depend on the future

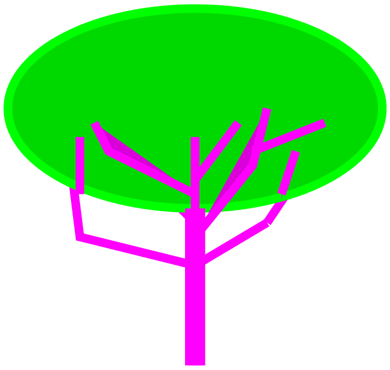
(II) f may not depend on the far past

Solution 1: Tree Automata

[Rabin 72]

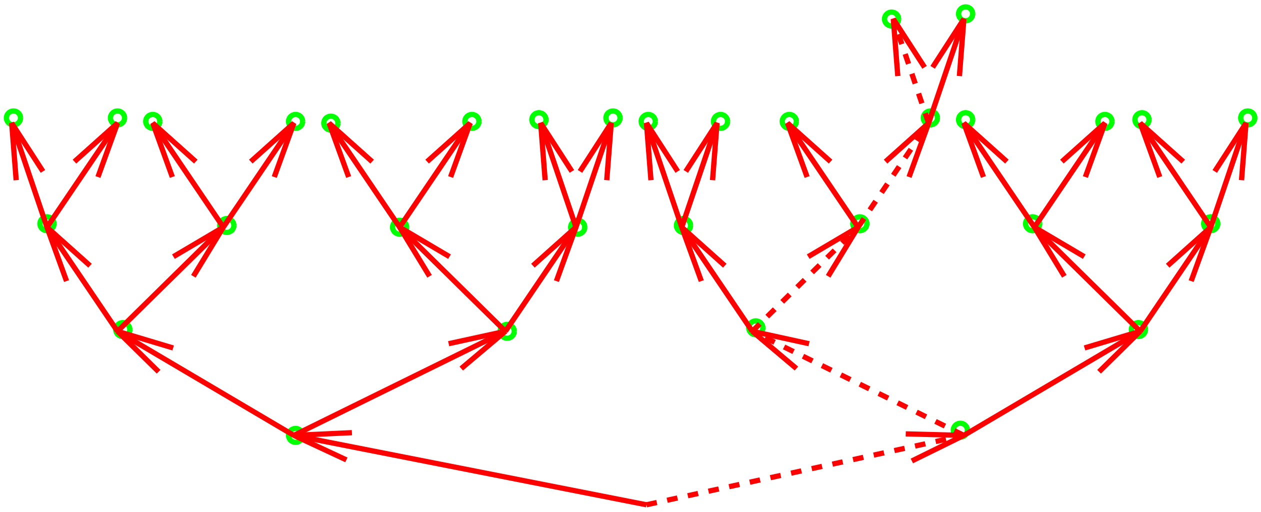
[Emerson-Jutla 88]

[Pnueli-Rosner 89]



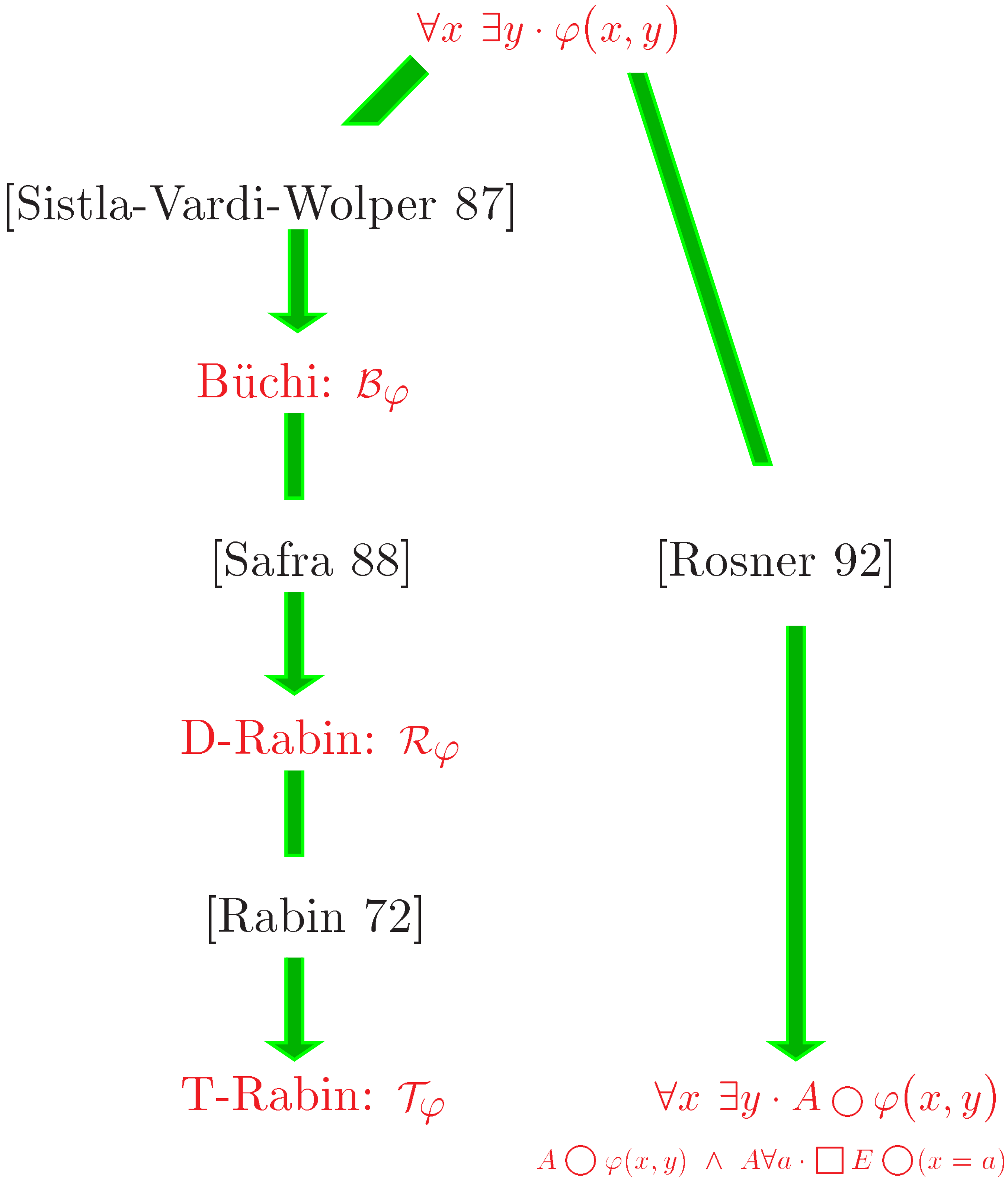
X-player chooses branch

Y-player chooses labeling



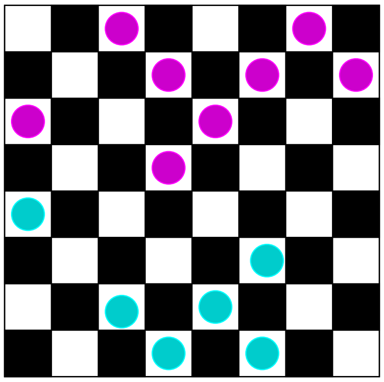
- Realizability = Non-emptiness
- Extract deterministic transducer from model
- Complexity: 2EXPTIME

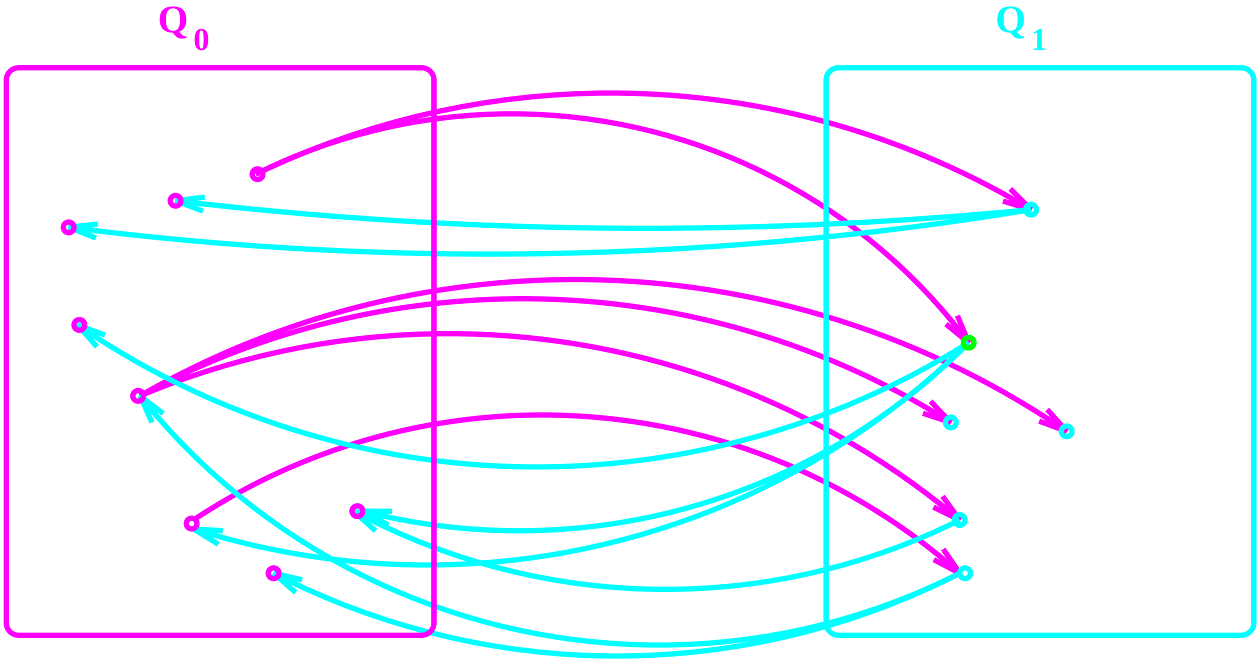
Specification $\mapsto$ Tree Automaton



Solution 2: Game Automata

[Gale-Stewart 53]
[Büchi-Landweber 69]
[McNaughton 93]
[Thomas 95][Lescow 95]





Specification $\subseteq (Q_0Q_1)^\omega$

[Davis 64], [Martin 75]:

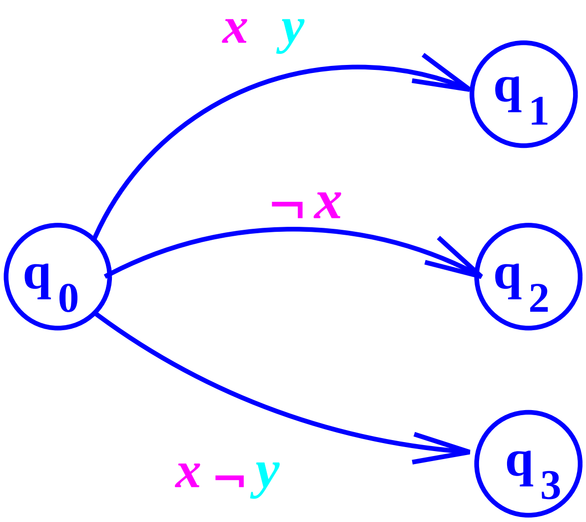
Sequential games $\subset$ Borel games

are determined

Specification $\mapsto$ Game

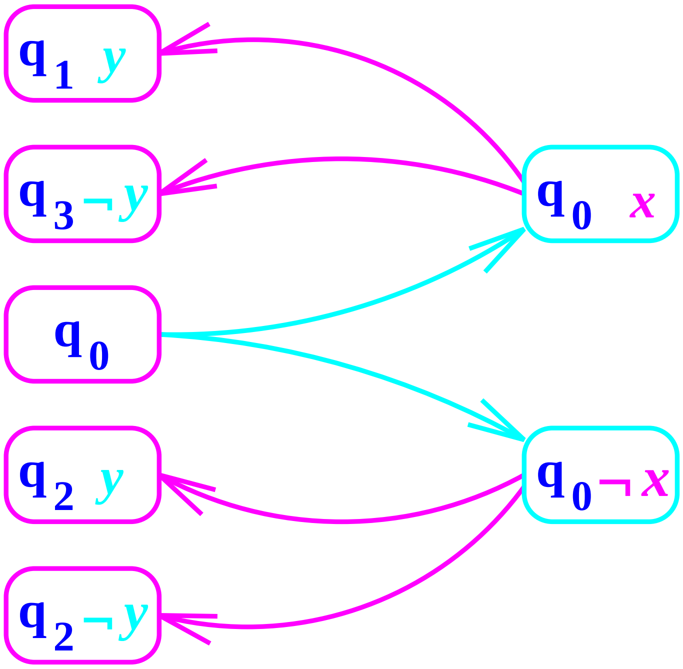
$\forall x \exists y \cdot \varphi(x, y)$

[Sistla-Vardi-Wolper 87]
[Safra 88]



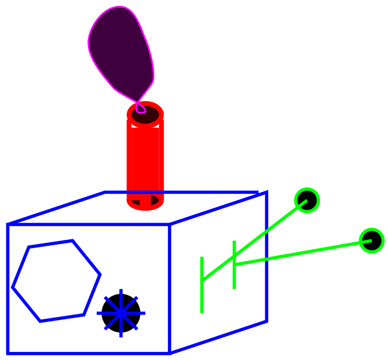
D-Automaton $\mathcal{R}_\varphi$

Game :



Solution 3: Control Synthesis

[Ramadge-Wonham 87]
[Thistle-Wonham 94]
[Maler-Pnueli-Sifakis 95]



Discrete Event System $\mathcal{P}$:

Σ	controllable actions
Q	system states
q_0	initial state
$\delta_{\mathcal{P}} : Q \times \Sigma \to 2^Q$	transition function
$L : Q \to 2^{\mathrm{Prop}}$	state labeling

Find controller

$$\mathcal{C} = \langle M,\ m_0,\ \delta_{\mathcal{C}} : M \to \Sigma \times M \rangle$$

such that

$$\mathcal{C} \times \mathcal{P} \models \varphi$$

Specification $\mapsto$ Controller

φ

$\mathcal{R}_\varphi$ D-Rabin

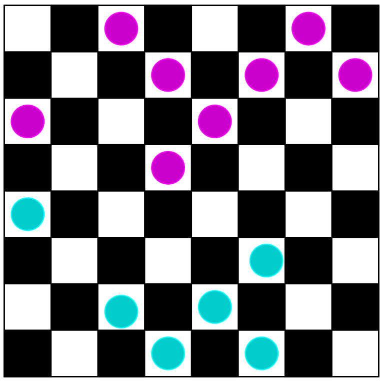
$\mathcal{P}$ Plant

$\mathcal{R}_\varphi \times \mathcal{P}$ Specification plant

[Thistle-Wonham 94]

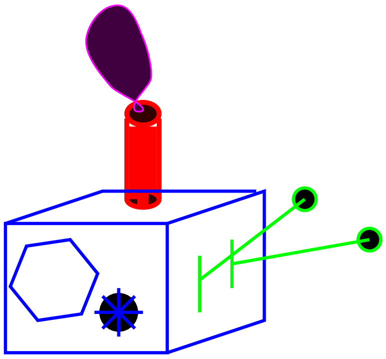
$\mathcal{C}$ Controller

Game $\mapsto$ Controller



$$\langle Q_0,\ Q_1,\ q_0,\ E_0,\ E_1,\ \Omega\rangle$$

$$\Big\Downarrow$$



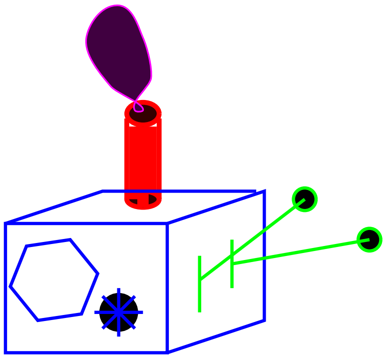
$$\langle Q,\ q_0,\ \Sigma,\ \delta,\ \Omega\rangle$$

$$Q = Q_0$$

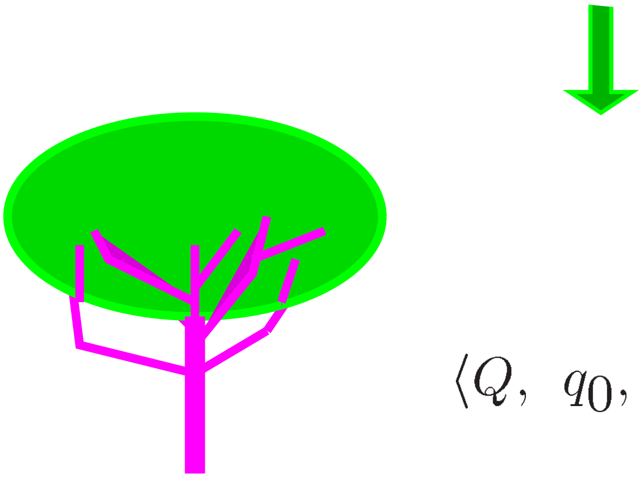
$$\Sigma = Q_1$$

$$\delta(q,a) = \{q' \in Q_0 \ :\ (q,a) \in E_0, (a,q') \in E_1\}$$

Controller $\mapsto$ Tree Automaton



$$\langle Q,\ q_0,\ \Sigma,\ \delta : Q \times \Sigma \to 2^Q,\ \Omega \rangle$$



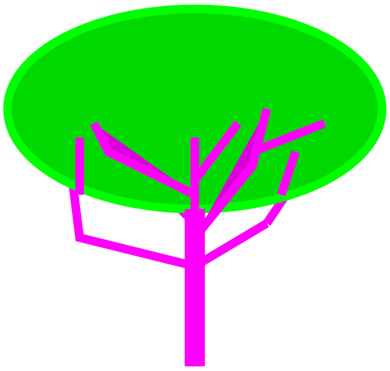
$$\langle Q,\ q_0,\ \Sigma,\ \delta_{\mathcal{T}} : Q \times \Sigma \to Q^*,\ \Omega \rangle$$

$$\delta(q,a) = \{q_0, \ldots, q_n\}$$

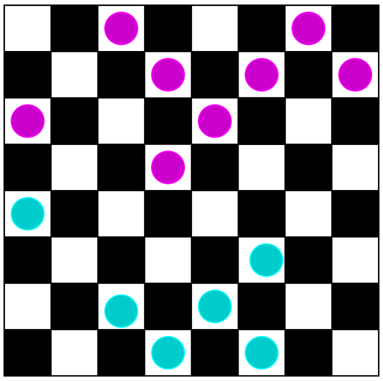
$$\Downarrow$$

$$\delta_{\mathcal{T}}(q,a) = \langle q_0, \ldots, q_n \rangle$$

Tree Automaton $\mapsto$ Game



$$\langle Q,\ q_0,\ \Sigma,\ \delta_{\mathcal{T}} : Q \times \Sigma \to Q^*,\ \Omega\rangle$$



$$\langle Q_0,\ Q_1,\ q_0,\ E_0,\ E_1,\ \Omega\rangle$$

$$Q_0 = Q$$

$$Q_1 = Q \times \Sigma$$

$$E_0 = \{(q, (q, a))\ :\ a \in \Sigma\}$$

$$E_1 = \{(((q, a), q')\ :\ \langle \ldots, q', \ldots\rangle = \delta_{\mathcal{T}}(q, a)\}$$

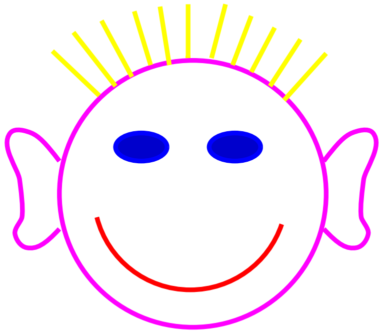
Complexity

	controller
Size of	transducer
	winning strategy

depends on acceptance condition

[Lescow 95 (CAV)] summarizes:

- Rabin - No memory needed
- Muller chain - Polynomial size
- Streett, Muller - Exponential size



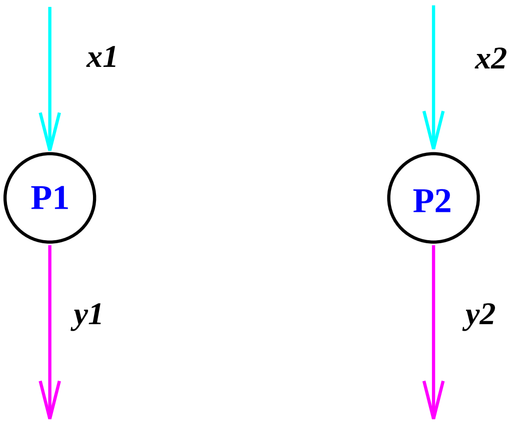
finite memory

Other Works on Complexity

[Rabin 69]	(LAR)
[Trakhtenbrot-Barzdin 70]	Monadic solvability
[Gurevich-Harrington 82]	LAR
[Klarlund 92]	Rabin-measures
[Nerode-Yakhnis-Yakhis 92]	Games
[Zeitman 94]	LAR
[Muller-Schupp 95]	IAR
[Krishnan-Puri-] [Brayton-Varaiya 95]	Rabin Index+ Chain automata
[Buhrke-Lescow-Vöge 96]	Streett+ Rabin chain

Multiprocessor synthesis

[Pnueli-Rosner 90]



- Hierarchical architectures

  NONELEMENTARY DECIDABLE

- Finite state

  UNDECIDABLE

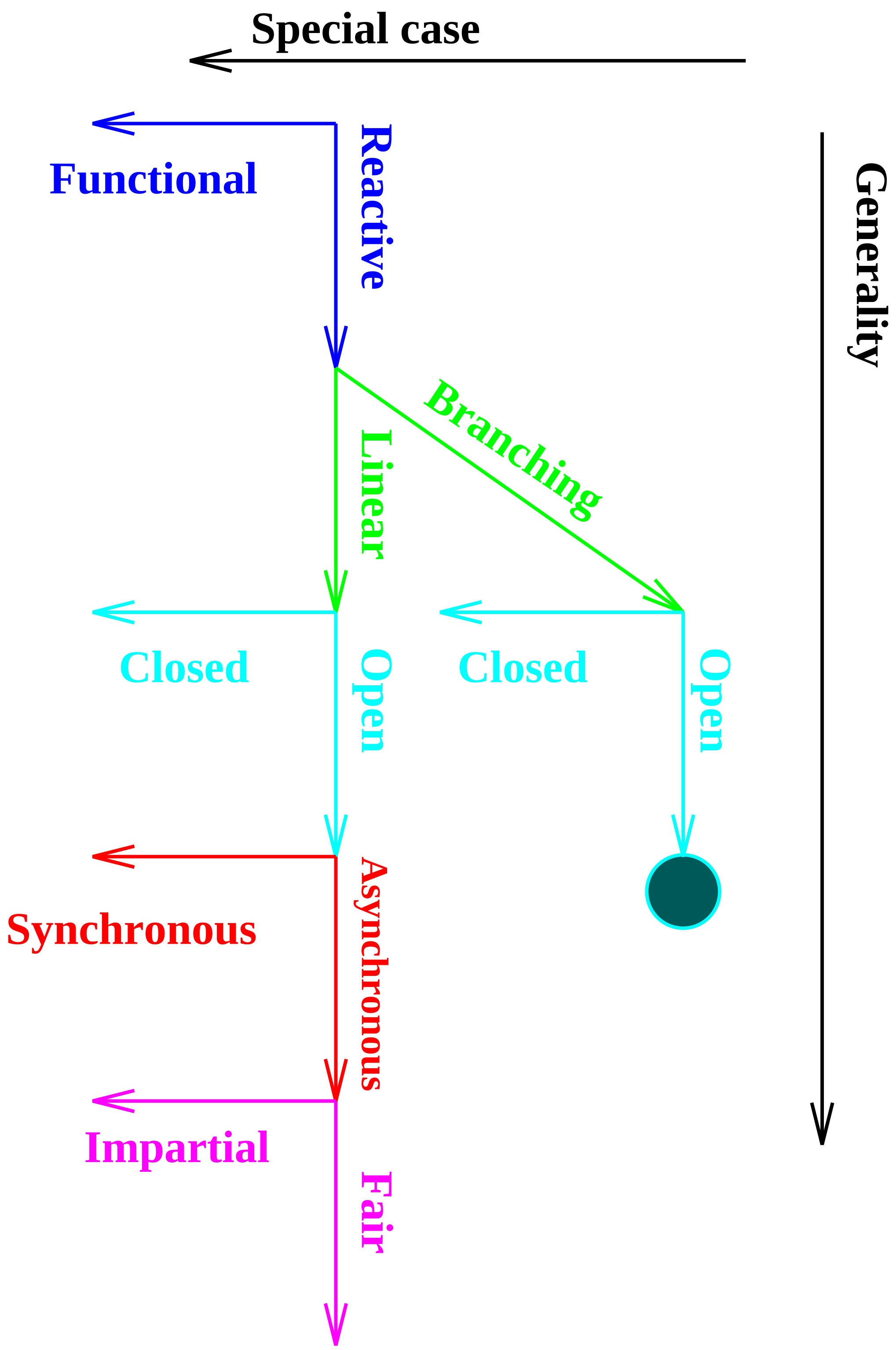
- Unrestricted

  HIGLY UNDECIDABLE

- Incomplete approach: Decomposition

  NONELEMENTARY DECIDABLE

Road Map



Special case
Generality
Functional
Reactive
Branching
Linear
Closed
Open
Closed
Open
Synchronous
Asynchronous
Impartial
Fair

Branches of the Church

Control plant to satisfy CTL specification.

State labeling CTL model-checking [Clarke-Emerson 81]

$\Downarrow$ adapted to

CTL control synthesis [Antoniotti 95,96]

Handles subset of CTL.

No memory allowed.

NP-complete for full CTL

Automata Theory for CTL

Tools developed for

- Alternating automata
  [Bernholtz 95] [Vardi 95]
- EQCTL
  [Kupferman 95]
- Module checking
  [Kupferman-Vardi 96 (CAV)]

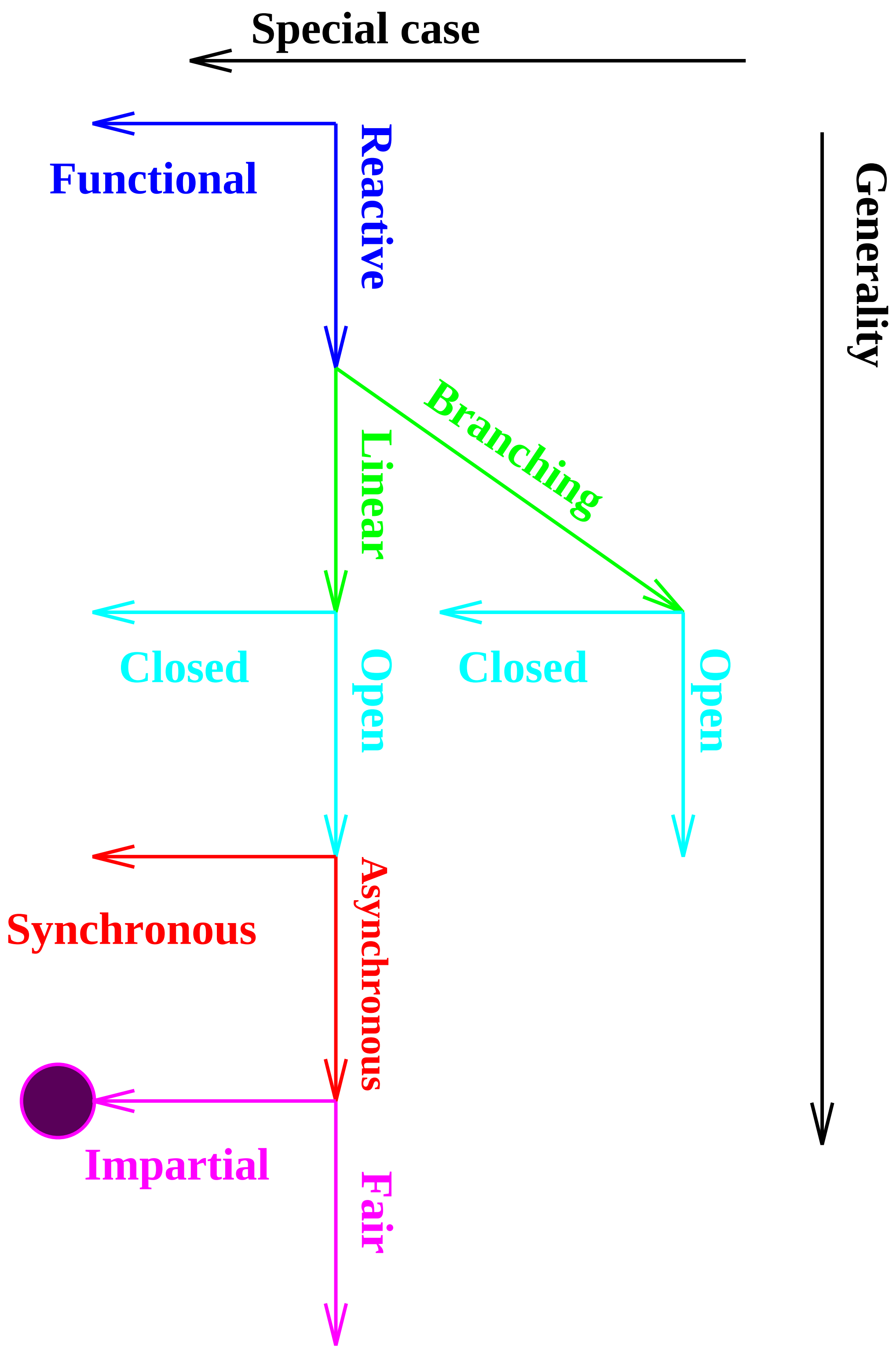
yield:

- CTL control (no memory allowed) is
  NP-complete
- CTL control (with memory) is
  EXPTIME complete
- CTL* control is
  2EXPTIME complete

Kupferman = Bernholtz

$\neg\Box(\text{Orna} = \text{Orna})$

Road Map


Special case
Functional
Reactive
Generality
Branching
Linear
Closed
Open
Closed
Open
Synchronous
Asynchronous
Impartial
Fair

Asynchronous Synthesis

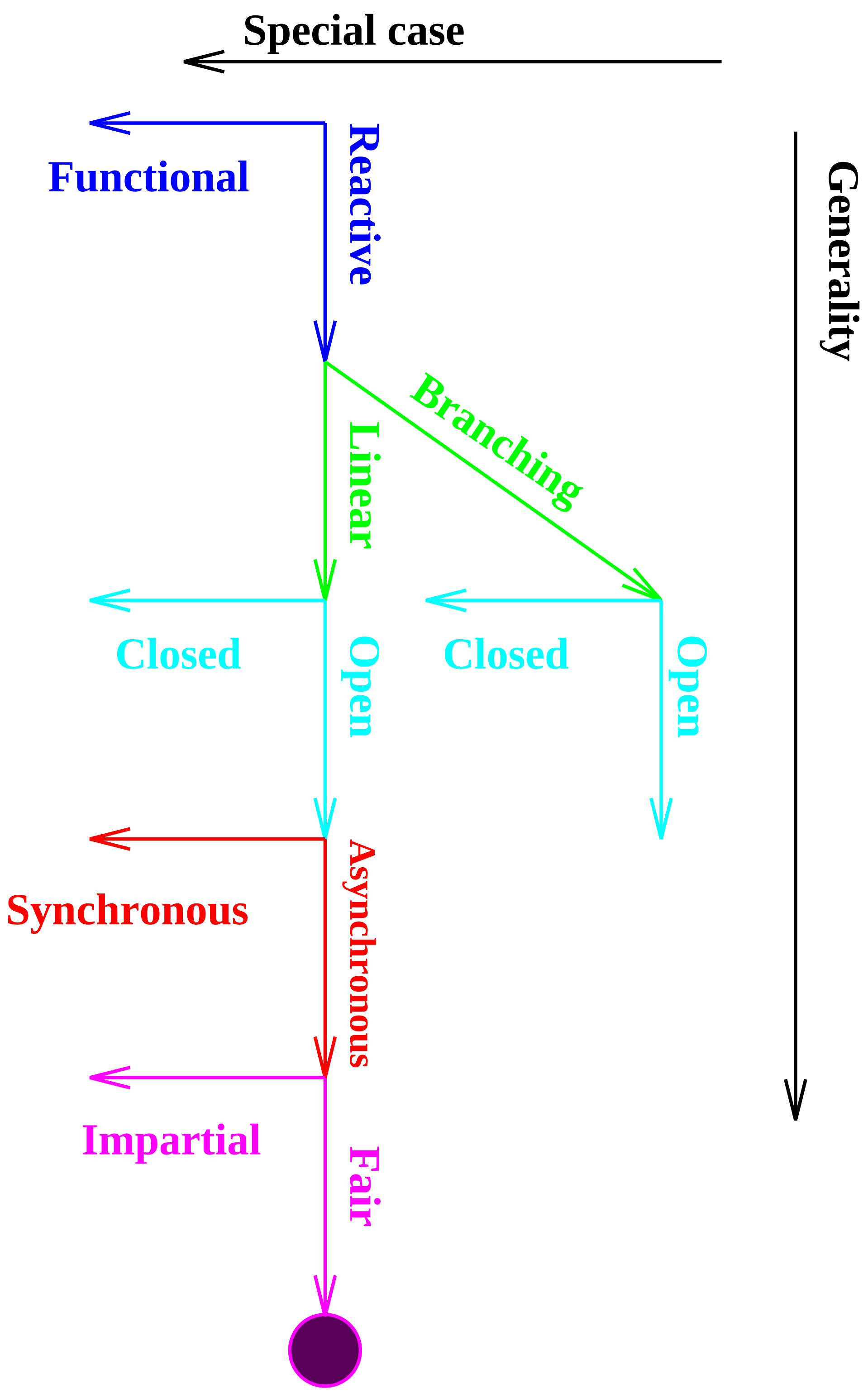
A third player: the scheduler.

- Environment and module interleave.
- Synthesized module only sees trace where it is scheduled.
- Scheduler is impartial: $\Box\Diamond$schedule.

[Pnueli-Rosner 89]
    Logical encoding
[Abadi-Lamport-Wolper 89]
    Failure traces
[Wong Toi-Dill 91]
    Flexible framework

2EXPTIME

Road Map



Special case
Functional
Reactive
Generality
Branching
Linear
Closed
Open
Closed
Open
Synchronous
Asynchronous
Impartial
Fair

Asynchronous Synthesis under Fairness

Scheduling under fairness assumptions:

- Just:

$$\Diamond \Box \text{Enabled} \rightarrow \Box \Diamond \text{schedule}$$

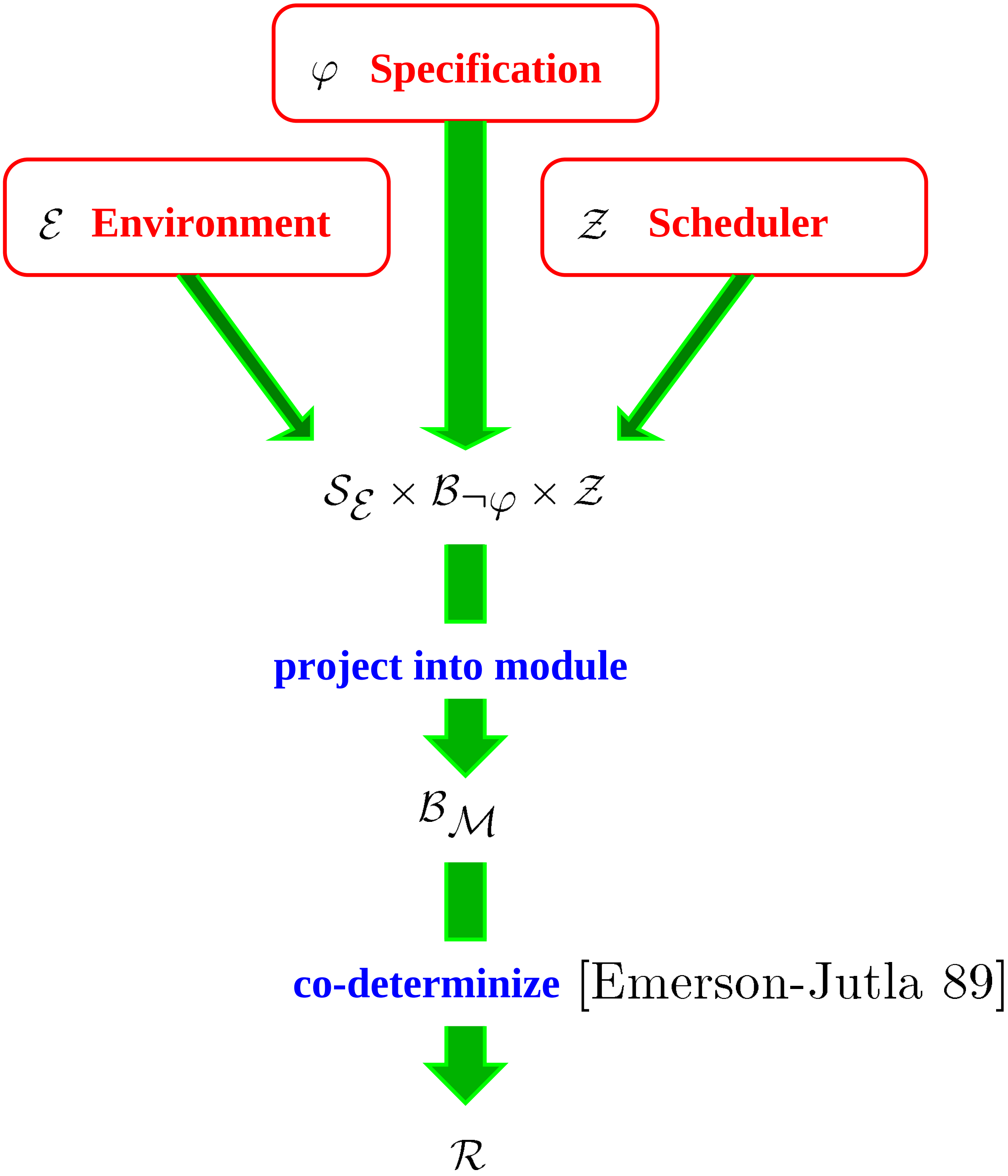
- Compassionate:

$$\Box \Diamond \text{Enabled} \rightarrow \Box \Diamond \text{schedule}$$

[Anuchitanukul-Manna 94]
Tableau based, ETL
[Vardi 95]
Automata based
2EXPTIME-complete in $\mathcal{B}$

3EXPTIME needed from LTL specification

General Modular Synthesis
[Anuchitanukul 95]



Real-time

Discrete control synthesis for real-time systems

[Wong Toi-Dill-Hoffmann 91,92]

Discretize region automata

[Maler-Pnueli-Sifakis-Asarin 95,96]

Controllable states from fixpoints

Real-time plant:

Σ	controllable actions
Q	system states
X	set of clocks
$F(X)$	reset functions on clocks
$I : Q \to \mathcal{H}_k^{\cap}$	state region invariants

$$R \quad \subseteq \quad Q \times \Sigma \times \mathcal{H}_k^{\cap} \times F(X) \times Q$$

Extract simple real-time controller:

$$C : Q \times X \to \Sigma \cup \{\text{clock}\}$$

Open Theoretical Problems

- Completing the branching time path
- Beyond $F_{\sigma\delta} \cap G_{\delta\sigma}$
  [Büchi 83]
- Games on pushdown automata
  [Thomas 95]
- More powerful real-time synthesis
  [Trachtenbrot 95]
- Synthesis for infinite state systems
- Proof-theoretical isomorphism for reactive synthesis
- Maximal synthesis for open systems
  [Anuchitanukul 95]

Open Practical Problems

From Toy examples
towards “Real” applications

Cats and mice
Mutual exclusion
[Ramadge-Wonham 89]
Semiconductor Manufacturing
[Wong Toi-Hoffmann 92]
Control of walking robot’s legs
Assembly line control
[Antoniotti 95]

Challenges

- More robots
- Highway control
- Flight control - a hybrid system
- Protocols in operating systems
- Bus protocols

References

Functional Synthesis work [BD77] [How80] [MW80] [Coq85] [CAB+86] [CH88] [HN88] [Smi90] [MT91] [Abr90]

Closed system synthesis [MW81] [CE81] [EC82] [MW84] [Anu95] [AE89]

Open synchronous [Chu62]

Trees [Rab72] [EJ88] [PR89b] [SVW87] [Saf88] [R.R92]

Games [GS53] [BL69] [McN93] [Les95, Tho95] [Dav64] [Mar75]

Control [RW89] [TW94a, TW94b] [MPS95] [OW90] [LW95]

Complexity [Les95] [Rab69] [GH82] [PR90] [Kla92] [NYY92] [YY93] [TL93] [KK91] [Tho94] [Zei94] [MS95] [KPBV95] [BLV96]

Branching [Ant95] [Kup95b] [Kup95a] [KV96] [Var96]

Asynchronous [PR89a] [ALW89] [DW91]

Fair [AM94] [Var95]

Real time [WH92] [MPS95] [AMP96]

Various [Büc83] [EJ88] [EJ89, EJ91] [EH93]

Bibliography

References

[Abr90] S. Abramsky. Computational Interpretations of Linear Logic. Technical report, Imperial College London, October 30 1990.

[AE89] P. C. Attie and E. A. Emerson. Synthesis of concurrent systems with many similar Sequential Processes. In **Proc. 16th ACM Symp. Princ. of Prog. Lang.**, 1989.

[ALW89] M. Abadi, L. Lamport, and P. Wolper. Realizable and unrealizable concurrent program specifications. In **Proc. 16th Int. Colloq. Aut. Lang. Prog.**, volume 372 of **LNCS**, pages 1–17. Springer-Verlag, 1989.

[AM94] A. Anuchitanukul and Z. Manna. Realizability and synthesis of reactive modules. In D.L. Dill, editor, **Proc.** 6^{th} **Intl. Conference on Computer Aided Verification**, volume 818 of **LNCS**, pages 156–168. Springer-Verlag, 1994.

[AMP96] E. Asarin, O. Maler, and A. Pnueli. Symbolic Controller Synthesis for Discrete and Timed Systems. In P. Antsaklis et al., editor, **Hybrid Systems II**, number 999 in LNCS, pages 1–20. Springer-Verlag, 1996.

[Ant95] M. Antoniotti. **Synthesis and Verification of Discrete Controllers for Robotics and Manufacturing Devices with Temporal Logic and the Control-D System**. PhD thesis, New York University, 1995. Available via http://www.icsi.berkeley.edu/ marcoxa.

[Anu95] A. Anuchitanukul. **Synthesis of Reactive Programs**. PhD thesis, Stanford University, 1995.

[BD77] R. M. Burstall and J. Darlington. A tranformation system for developing recursive programs. **J. Assoc. Comput. Mach.**, 24:44–67, 1977.

[BL69] J.R. Büchi and L. H. Landweber. Solving sequential conditions by finite-state strategies. **Trans. Amer. Math. Soc.**, 138:295–311, 1969.

[BLV96] N. Buhrke, H. Lescow, and J. Vöge. Strategy construction in infintie games with Streett and Rabin chain winning conditions. In **TACAS 96**, page 18, 1996.

[Büc83] J.R. Büchi. State-strategies for games in $F_{\sigma\delta} \cap G_{\delta\sigma}$. **J. Symb. Logic**, 48:1171–1198, 1983.

[CAB+86] R.L. Constable, S.F. Allen, H.M. Bromley, W.R. Cleaveland, J.F. Cremer, R.W. Harper, D.J. Howe, T.B. Knoblock, N.P. Mendler, P. Panangaden, J.T. Sasaki, and S.F. Smith. **Implementing Mathematics with the**

Nuprl Proof Development System. Prentice-Hall, Englewood Cliffs, NJ, 1986.

[CE81] E.M. Clarke and E.A. Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In **Proc. IBM Workshop on Logics of Programs**, volume 131 of **LNCS**, pages 52–71. Springer-Verlag, 1981.

[CH88] Th. Coquand and G. Huet. The Calculus of Constructions. **Information and Computation**, 76(2/3), 1988.

[Chu62] A. Church. Logic, Arithmetic, and Automata. In **International Symposium of Mathematicians**, pages 23–35, 1962.

[Coq85] Th. Coquand. **Une Théorie des Constructions**. PhD thesis, Université Paris 7, 1985.

[Dav64] M. Davis. Infintie games of perfect information. In **Advances in game theory**, pages 85–101. Princeton Univ. Press, Princeton N. J., 1964.

[DW91] D.L. Dill and H. Wong-Toi. Synthesizing processes and schedulers from temporal specifications. In R.P. Kurshan and E.M. Clarke, editors, **CAV 90: Automatic Verification Methods for Finite-state Systems**, LNCS. Springer-Verlag, 1991.

[EC82] E.A. Emerson and E.M. Clarke. Using branching time temporal logic to synthesize synchronization skeletons. **Sci. Comp. Prog.**, 2(3):241–266, 1982.

[EH93] J. Engelfriet and H. J. Hoogeboom. X-automata on ω-words. **Theor. Comp. Sci.**, 110(1):1–51, 1993.

[EJ88] E. A. Emerson and C. S. Jutla. The Complexity of Tree Automata and Logics of Programs (Extended Abstract). In **FOCS 88: Foundations of Computer Science**, 1988.

[EJ89] E.A. Emerson and C.S. Jutla. On simultaneously determinizing and complementing ω-automata. In **Proc. 4th IEEE Symp. Logic in Comp. Sci.**, 1989.

[EJ91] E.A. Emerson and C.S. Jutla. Tree automata, mu-calculus and determinacy. In **Proc. 6th IEEE Symp. Logic in Comp. Sci.**, 1991.

[GH82] Y. Gurevich and L.A. Harrington. Automata, trees and games. In **Proc. 14th ACM Symp. Theory of Comp.**, pages 60–65, 1982.

[GS53] D. Gale and F. M. Stewart. Infintie games with perfect information. **Annals of Mathematical Studies**, 28:245–266, 1953.

[HN88] S. Hayashi and H. Nakano. **PX: A Computational Logic**. The MIT Press, Cambridge, Mass, 1988.

[How80] W. A. Howard. The formuae-as-types notion of constructions. In Seldin J.P and Hindley J. R., editors, **to H.B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism.** Academic Press, 1980.

[KK91] N. Klarlund and D. Kozen. Rabin measures and their applications to fairness and automata theory. In **Proc. 6th IEEE Symp. Logic in Comp. Sci.**, pages 256–265, 1991.

[Kla92] N. Klarlund. Progress measures, immediate determinacy, and a subset construction for tree automata. In **Proc. 7th IEEE Symp. Logic in Comp. Sci.**, pages 382–393, 1992.

[KPBV95] S.C. Krishnan, A. Puri, R.K. Brayton, and P.P Varaiya. The Rabin Index and Chain automata, with applications to automomata and games. In **Proc.** 7^{th} **Intl. Conference on Computer Aided Verification**, pages 253–266, 1995.

[Kup95a] O. Kupferman. Augmenting Branching Temporal Logics with Existential Quantification over Atomic Propositions. In **Proc.** 7^{th} **Intl. Conference on Computer Aided Verification**, pages 325–338, 1995.

[Kup95b] O. Kupferman. **Model Checking for Branching-Time Temporal Logics**. PhD thesis, Technion Haifa, 1995.

[KV96] O. Kupferman and M. Vardi. Module checking. In **Proc.** 8^{th} **Intl. Conference on Computer Aided Verification**, page 12, 1996. Unpublished manuscript.

[Les95] H. Lescow. On polynomial-size programs winning finite-state games. In **Proc.** 7^{th} **Intl. Conference on Computer Aided Verification**, pages 239–252, 1995.

[LW95] F. Lin and W. M. Wonham. Supervisory control of timed discrete-event systems under partial observation. **Transactions of the IEEE**, 40(3):558–562, 1995.

[Mar75] D.A. Martin. Borel determinacy. **Ann. Math.**, 102:363–371, 1975.

[McN93] R. McNaughton. Infinite games played on finite graphs. **Annals of Pure and Applied Logic**, 65:149–184, 1993.

[MPS95] O. Maler, A. Pnueli, and J. Sifakis. On the synthesis of diescrete controllers for timed systems (an extended abstract). In **Proc. of the** 12^{th} **Annual Symp. on Theoretical Aspects of Computer Science**, pages 229–242, 1995.

[MS95] D. E. Muller and P. E. Schupp. Simulating alternating tree automata by nondeterministic automata: New results and new proofs of the theorems by Rabin, McNaughton and Safra. **Theor. Comp. Sci.**, 141:69–107, 1995.

[MT91] G. Mints and E. Tyugu. The Programming System Priz. In J. Barzdins, D. Bjørner, editor, **Baltic Computer Science LNCS 502**, pages 1–17. Springer Verlag, 1991.

[MW80] Z. Manna and R.J. Waldinger. A deductive approach to program synthesis. **ACM Trans. Prog. Lang. Sys.**, 2:90–121, 1980.

[MW81] Z. Manna and P. Wolper. Synthesis of communicating processes from temporal logic specifications. In **Proc. IBM Workshop on Logics of Programs**, volume 131 of **LNCS**, pages 253–281, 1981.

[MW84] Z. Manna and P. Wolper. Synthesis of communicating processes from temporal logic specifications. **ACM Trans. Prog. Lang. Sys.**, 6(1):68–93, January 1984.

[NYY92] A. Nerode, A. Yakhnis, and V. Yakhnis. Concurrent Programs as Strategies in Games. In Y. Moschowakis, editor, **Logic From Computer Science**, pages 405–479. ??, 1992.

[OW90] J. S. Ostroff and W. M. Wonham. A framework for real-time discrete event control. **Transactions of the IEEE**, 35(4), 1990.

[PR89a] A. Pnueli and R. Rosner. On the synthesis of a reactive module. In **Proc. 16th ACM Symp. Princ. of Prog. Lang.**, pages 179–190, 1989.

[PR89b] A. Pnueli and R. Rosner. On the synthesis of an asynchronous reactive module. In **Proc. 16th Int. Colloq. Aut. Lang. Prog.**, volume 372 of **LNCS**, pages 652–671. Springer-Verlag, 1989.

[PR90] A. Pnueli and R. Rosner. Distributed reactive systems are hard to synthesize. In **Proc. 31th IEEE Symp. Found. of Comp. Sci.**, pages 746–757, 1990.

[Rab69] M.O. Rabin. Decidability of second order theories and automata on infinite trees. **Trans. Amer. Math. Soc.**, 141:1–35, 1969.

[Rab72] M.O. Rabin. **Automata on Infinite Objects and Church's Problem**, volume 13 of **Regional Conference Series in Mathematics**. Amer. Math. Soc., 1972.

[R.R92] R.Rosner. **Modular Synthesis of Reactive Systems**. PhD thesis, The Weizmann Institute of Science, 1992.

[RW89] P.J.G. Ramadge and W.M. Wonham. The control of discrete event systems. **Proc. of the IEEE on Control Theory**, 77:81–98, 1989.

[Saf88] S. Safra. On the complexity of ω-automata. In **Proc. 29th IEEE Symp. Found. of Comp. Sci.**, pages 319–327, 1988. An extended version to appear in *J. Comp. Sys. Sci.*

[Smi90] D.R. Smith. KIDS: a semiautomatic program-development system. **IEEE Trans. Software Engin.**, 16(9):1024–1043, September 1990.

[SVW87] A.P. Sistla, M.Y. Vardi, and P. Wolper. The complementation problem for Büchi automata with applications to temporal logic. **Theor. Comp. Sci.**, 49:217–237, 1987.

[Tho94] W. Thomas. Finite-State Strategies in Regular Infinite Games. In **LNCS vol 880**, 1994.

[Tho95] W. Thomas. On the synthesis of strategies in infinite games. In **Proc. of the 12^{th} Annual Symp. on Theoretical Aspects of Computer Science**, 1995.

[TL93] W. Thomas and H. Lescow. Logical Specifications of Infinite Computations. In **Rex: A Decade of Concurrency**, 1993.

[TW94a] J. G. Thistle and W. M. Wonham. Control of infinite behaviour of finite automata. **SIAM J. Control**, 32(1):1075–1097, 1994.

[TW94b] J. G. Thistle and W. M. Wonham. Supervision of infinite behaviour of discrete-event systems. **SIAM J. Control**, 32(1):1098–1113, 1994.

[Var95] M. Y. Vardi. An Automata-Theoretic Approach to Fair Realizability and Synthesis. In P. Wolper, editor, **Computer Aided Verification**, number 939 in LNCS. Springer-Verlag, 1995.

[Var96] M. Y. Vardi. Alternating automata and program verification. In J. v. Leuven, editor, **Current Trends in Computer Science**, number 1000 in LNCS. Springer-Verlag, 1996.

[WH92] H. Wong-Toi and G. Hoffmann. The control of dense real-time discrete event systems. Technical report, Stanford University, STAN-CS-92-1411, 1992.

[YY93] A. Yakhnis and V. Yakhnis. Gurevich-Harrington's games defined by finite automata. **Annals of Pure and Applied Logic**, 62:265–294, 1993.

[Zei94] S. Zeitman. Unforgettable Forgetful Determinacy. **Journal of Logic and Computation**, 4(3):273–283, 1994.