Trail Search with CRHS Equations

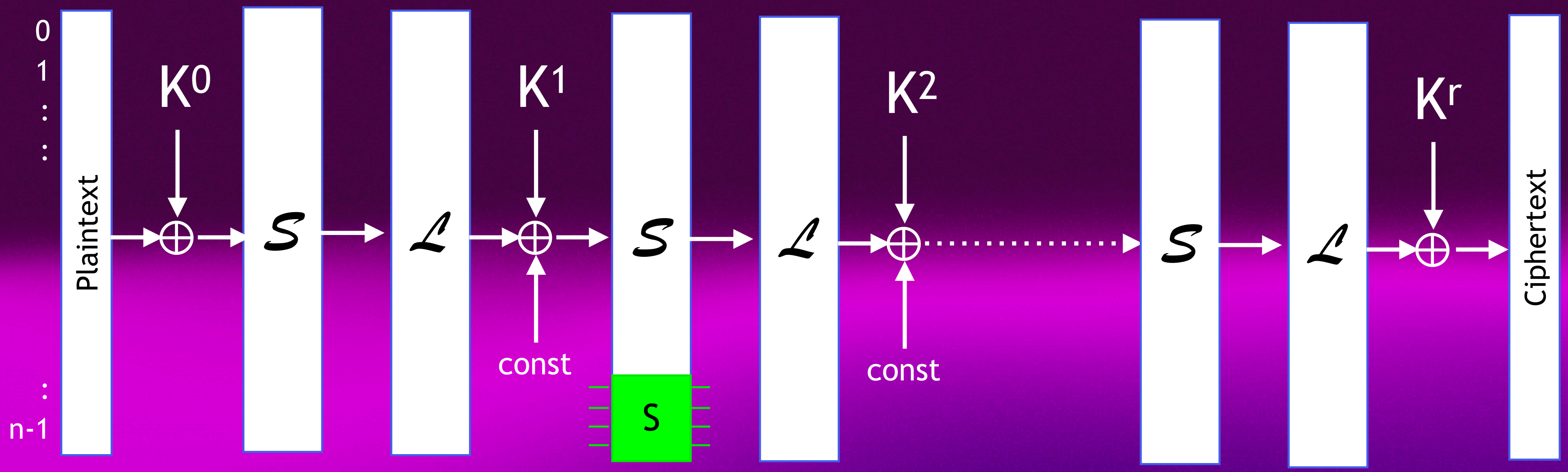
John Petter Indrøy and Håvard Raddum

https://eprint.iacr.org/2021/1329

Outline

- Finding good trails in block ciphers
- CRHS equations
- Using CRHS equations to find trails
- Results, Pathfinder and CryptaGraph

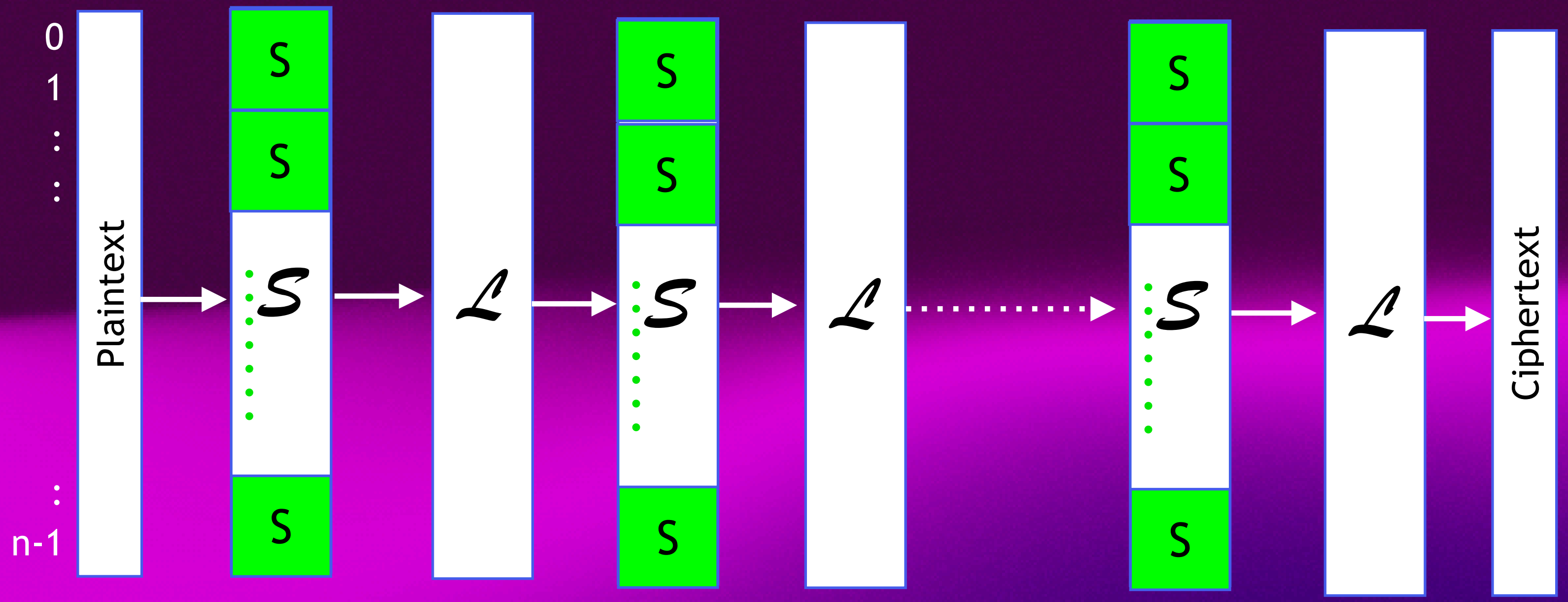
Classic block cipher design



Linear and differential attacks

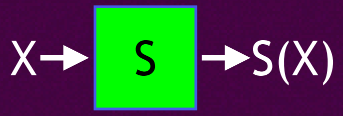
- Some of the oldest types of attacks (early 90's)
- Disregard addition of keys and constants in analysis
- Attack efficiency depends on interplay between $\mathcal{S}$ and $\mathcal{L}$
- New designs must prove resistance against linear and differential attacks

Cipher model



DDT and LAT

- S-box characterized by differential distribution table (DDT) and linear approximation table (LAT)



X	0	1	2	3	4	5	6	7
S(X)	2	5	1	0	6	4	3	7

$$DDT[\alpha][\beta] = |\{x \in \mathbb{F}_2^t \mid S(x) \oplus S(x \oplus \alpha) = \beta\}|$$

$$LC[\alpha][\beta] = |\{x \in \mathbb{F}_2^t \mid \langle x, \alpha \rangle = \langle S(x), \beta \rangle\}|$$

$$LAT[\alpha][\beta] = |2LC[\alpha, \beta] - 2^t|$$

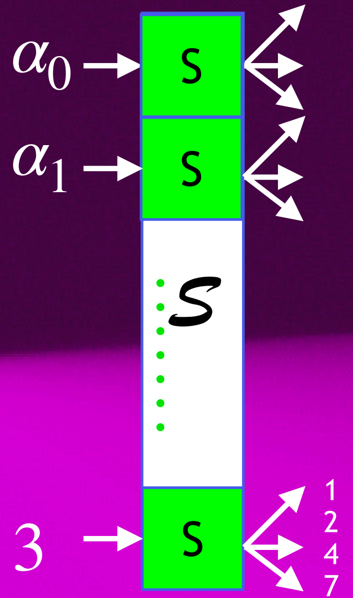
** DDT **

	0	1	2	3	4	5	6	7
0	8	0	0	0	0	0	0	0
1	0	2	2	0	2	0	0	2
2	0	0	0	4	0	4	0	0
3	0	2	2	0	2	0	0	2
4	0	2	2	0	2	0	0	2
5	0	0	0	4	0	0	4	0
6	0	2	2	0	2	0	0	2
7	0	0	0	0	0	4	4	0

** LAT **

	0	1	2	3	4	5	6	7
0	8	0	0	0	0	0	0	0
1	0	0	4	4	4	4	0	0
2	0	4	0	4	4	0	4	0
3	0	4	4	0	0	4	4	0
4	0	0	4	4	4	4	0	0
5	0	0	0	0	0	0	0	8
6	0	4	4	0	0	4	4	0
7	0	4	0	4	4	0	4	0

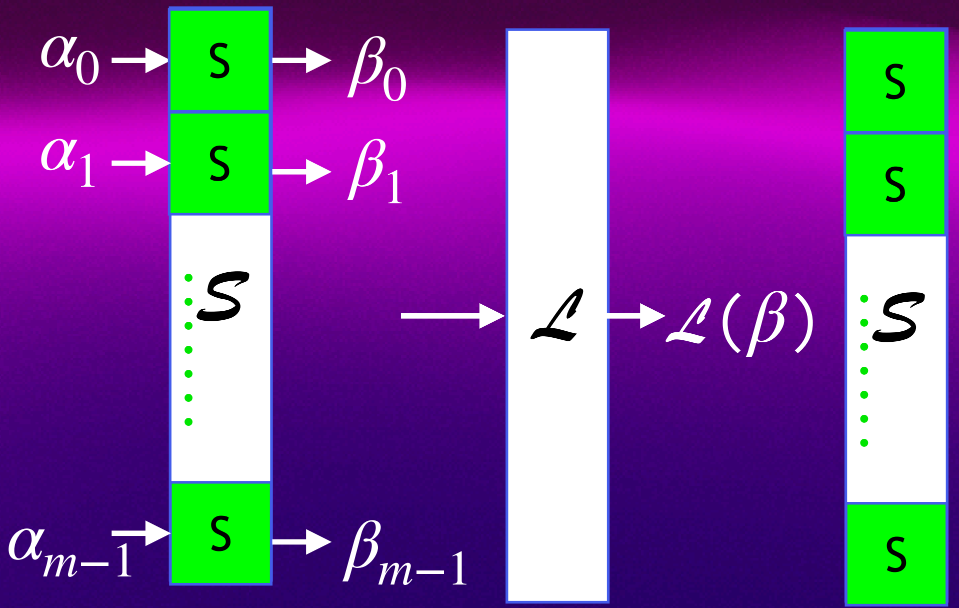
Starting a trail



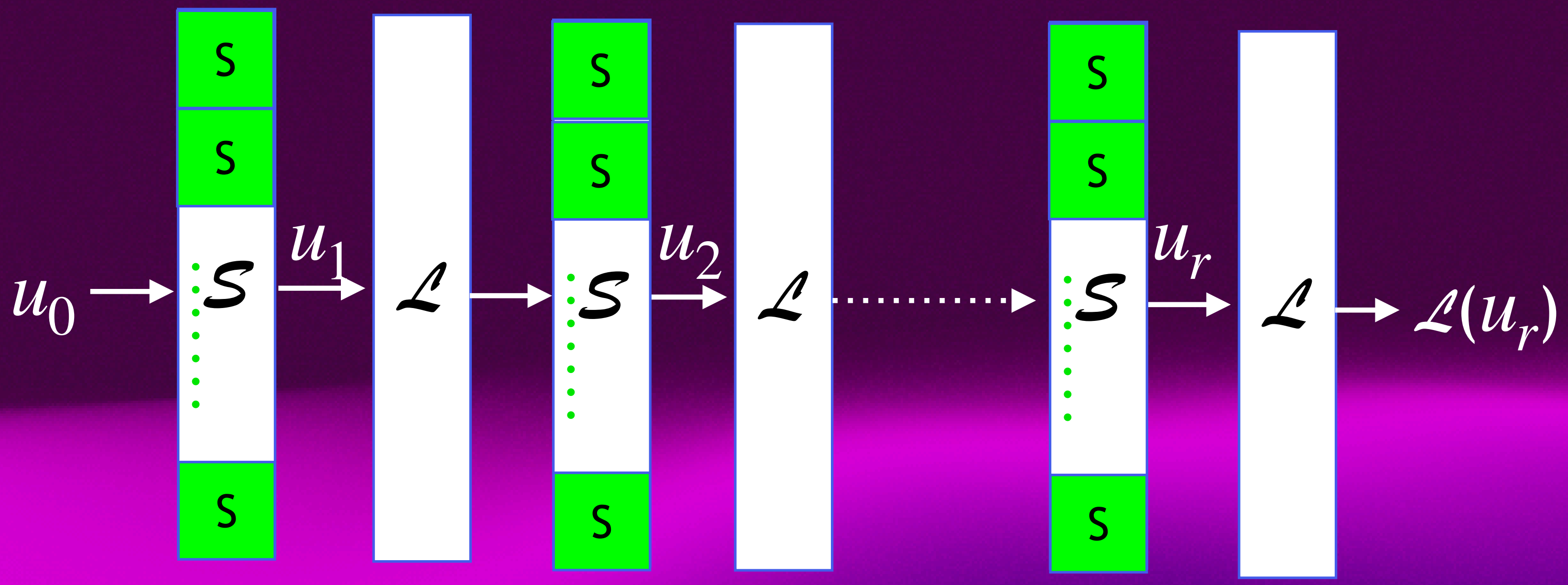
One input gives many possible outputs through $\mathcal{S}$

	0	1	2	3	4	5	6	7
0	8	0	0	0	0	0	0	0
1	0	2	2	0	2	0	0	2
2	0	0	0	4	0	4	0	0
3	0	2	2	0	2	0	0	2
4	0	2	2	0	2	0	0	2
5	0	0	0	4	0	0	4	0
6	0	2	2	0	2	0	0	2
7	0	0	0	0	0	4	4	0

Input to next $\mathcal{S}$ uniquely determined by output from previous $\mathcal{S}$



Complete trails

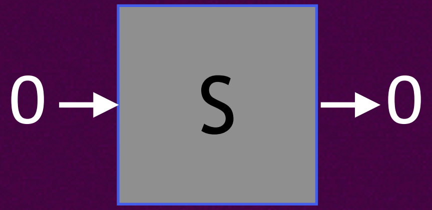


Trail: $\mathbf{u} = (u_0, \ldots, u_r)$ such that u_1 is possible output of u_0 and u_{i+1} is possible output from $\mathcal{L}(u_i)$ for $1 \leq i \leq r - 1$

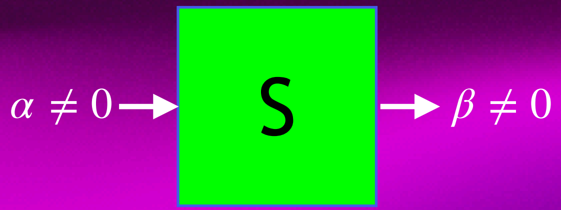
Hull: set of trails where all trails have the same u_0 and u_r

Active and passive S-boxes

Passive S-box

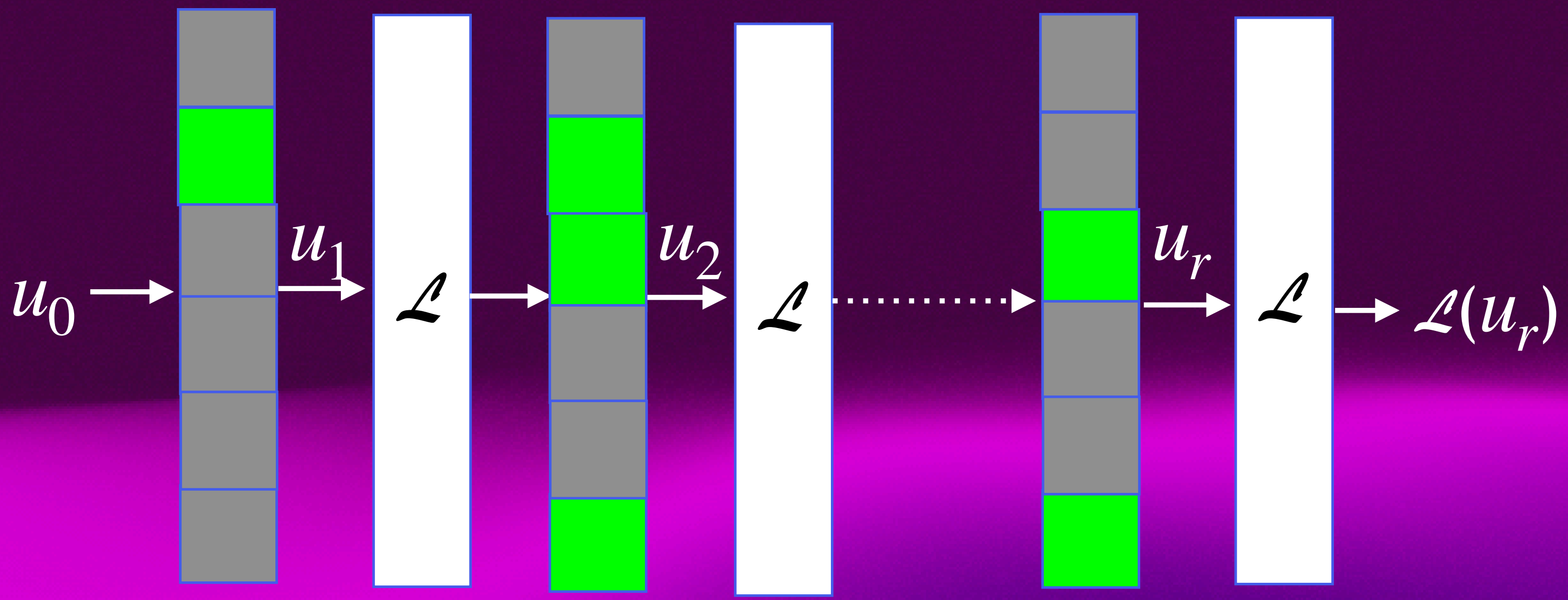


Active S-box



	0	1	2	3	4	5	6	7
0	8	0	0	0	0	0	0	0
1	0	2	2	0	2	0	0	2
2	0	0	0	4	0	4	0	0
3	0	2	2	0	2	0	0	2
4	0	2	2	0	2	0	0	2
5	0	0	0	4	0	0	4	0
6	0	2	2	0	2	0	0	2
7	0	0	0	0	0	4	4	0

Weight of trails



Active S-boxes contribute to weight of trail, $w(\mathbf{u})$, passive do not

Compexity of attacks $\approx 2^{w(\mathbf{u})}$

Core problem: Find valid trails with few active S-boxes

Methods for trail search

- Represent as MILP problem
- Use SAT or SMT solver
- Clever exhaustive search using tree structure with pruning
- Graph-based approach
  - [1] CryptaGraph, FSE 2018, https://eprint.iacr.org/2018/764

All of them have a problem when number of rounds increases

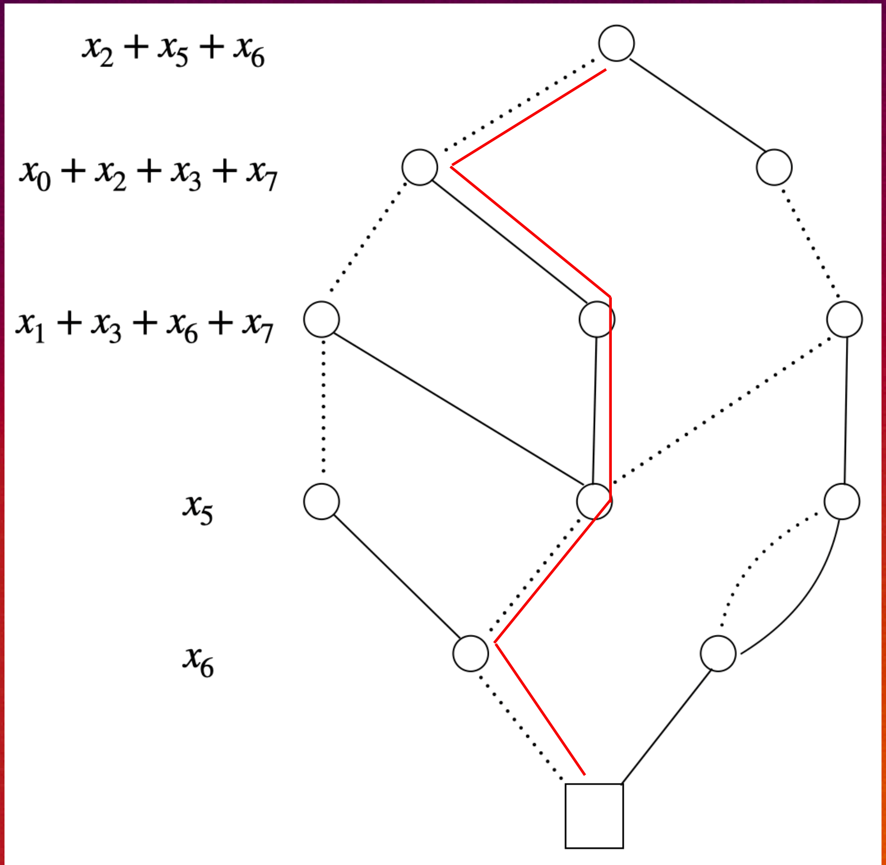
CRHS equations

– working with exponentially large sets

CRHS equation

- Graph with nodes arranged in horisontal levels
- One node on top level, one node on bottom level
- At most two outgoing edges from nodes: 0-edge and 1-edge
- Edges go from node on one level to node on level below
- Linear combinations of variables associated with levels

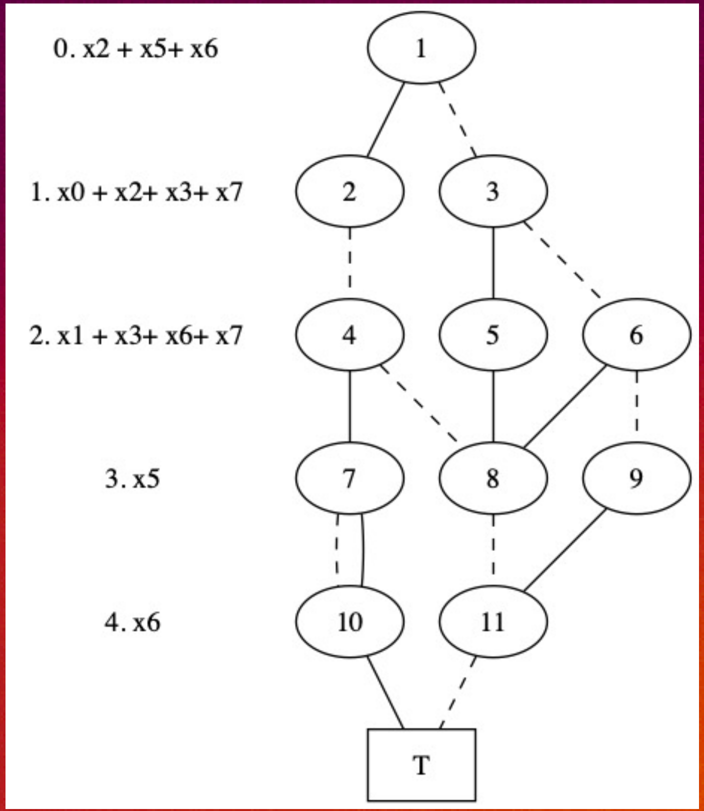
CRHS equation



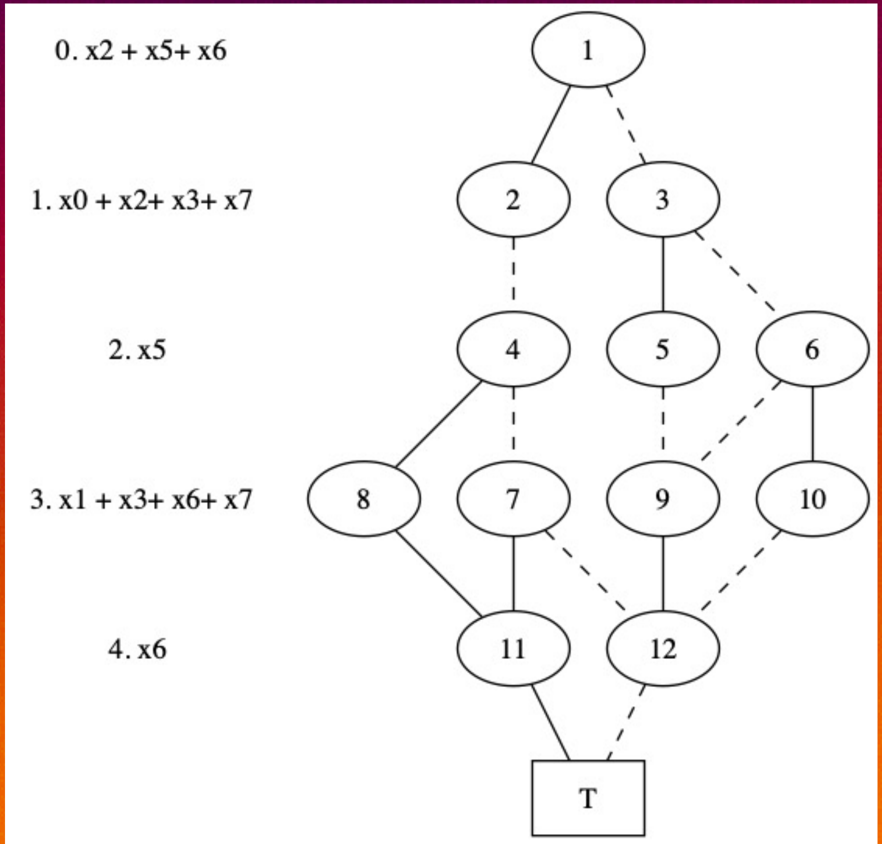
$$
\begin{aligned}
x_2 + x_5 + x_6 &= 0 \\
x_0 + x_2 + x_3 + x_7 &= 1 \\
x_1 + x_3 + x_6 + x_7 &= 1 \\
x_5 &= 0 \\
x_6 &= 0
\end{aligned}
$$

Solution set to CRHS equation:
union of solutions sets to $Ax = b$
for all b encoded as paths in graph

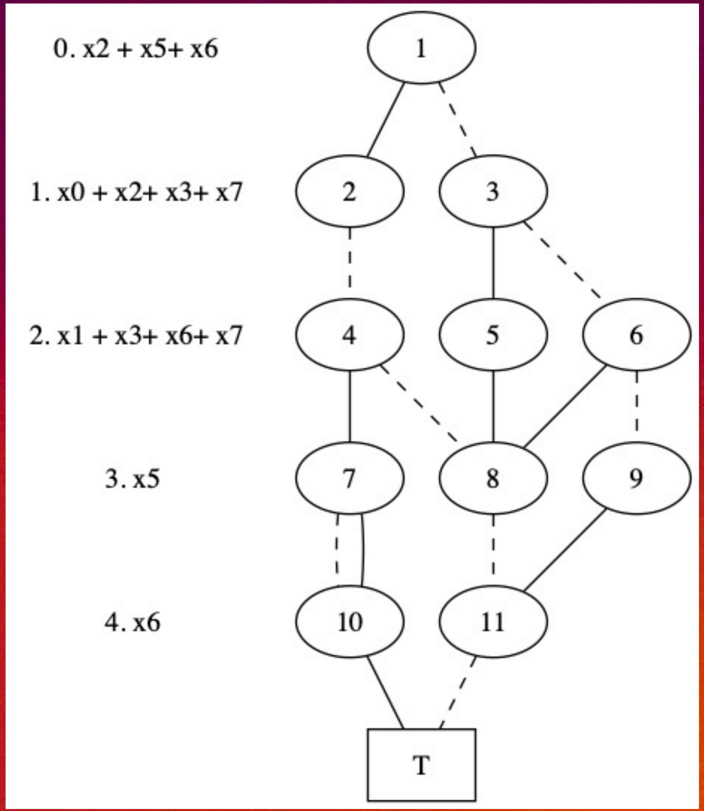
Operations on CRHS equation



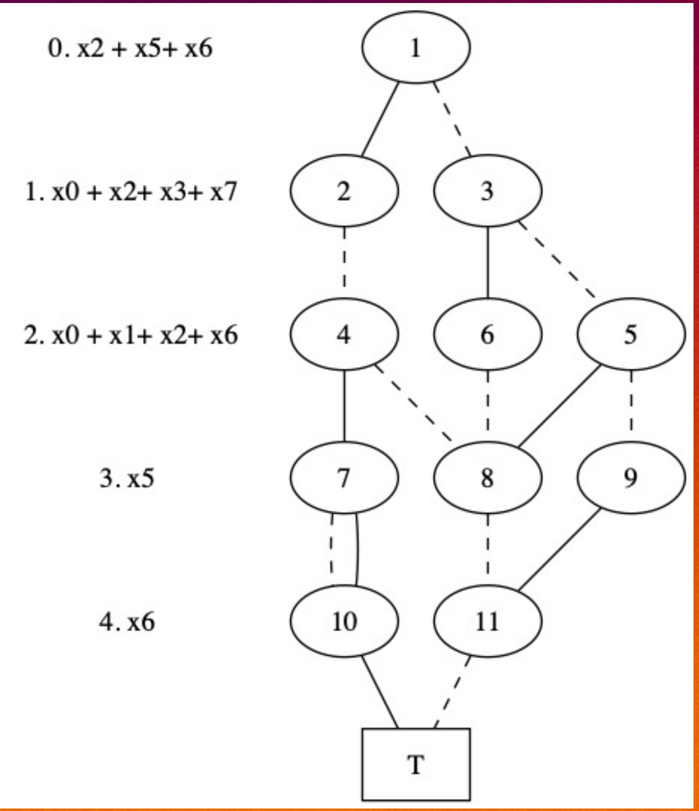
Swap two adjacent levels



Operations on CRHS equation

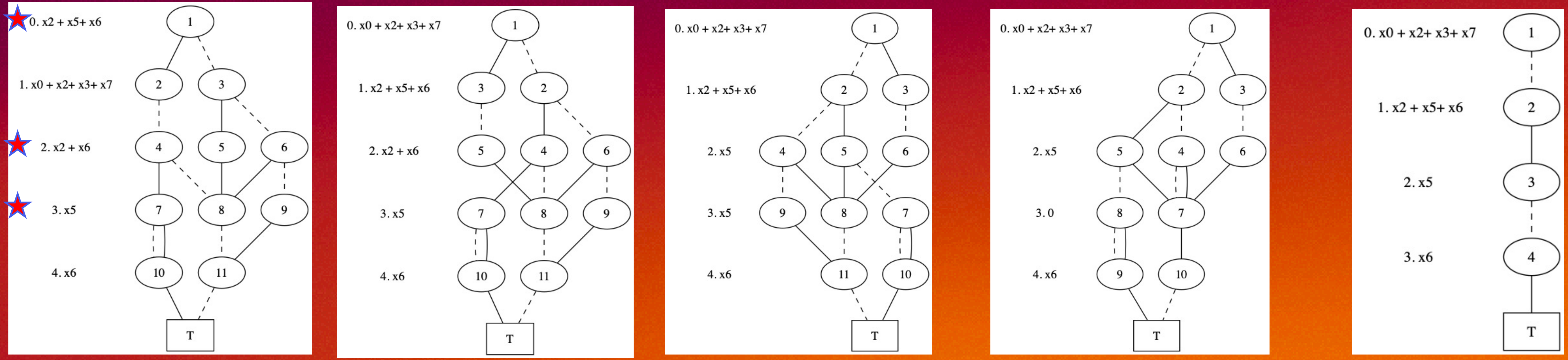


Add linear combination of one level onto linear combination on level below



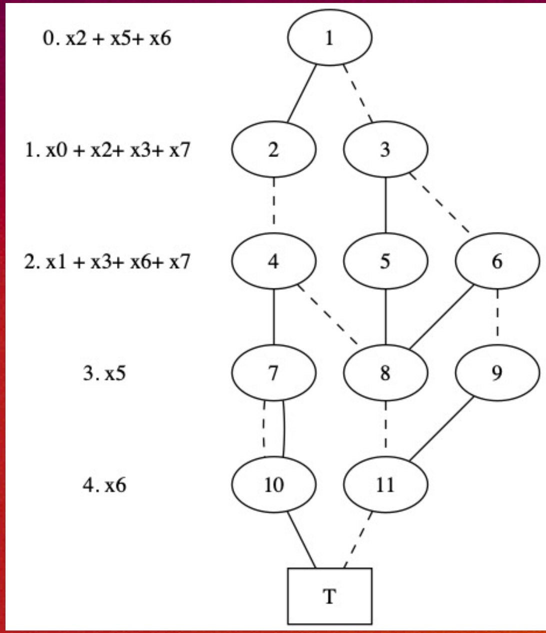
Linear absorption

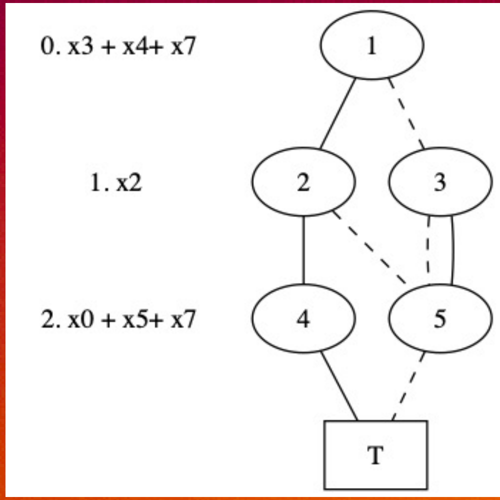
Linear dependencies among linear combinations can be removed

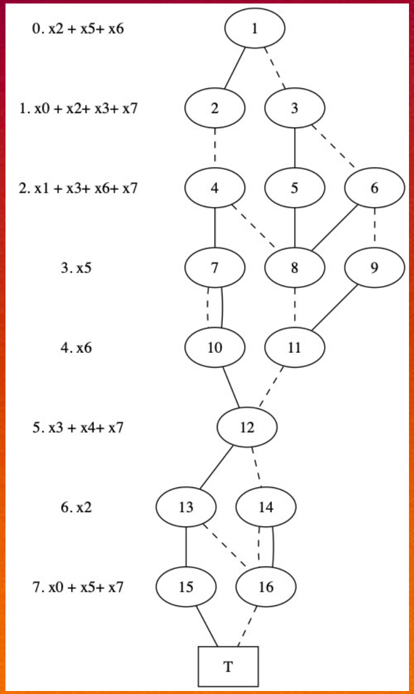


Joining CRHS equations

Two CRHS equations can easily be joined

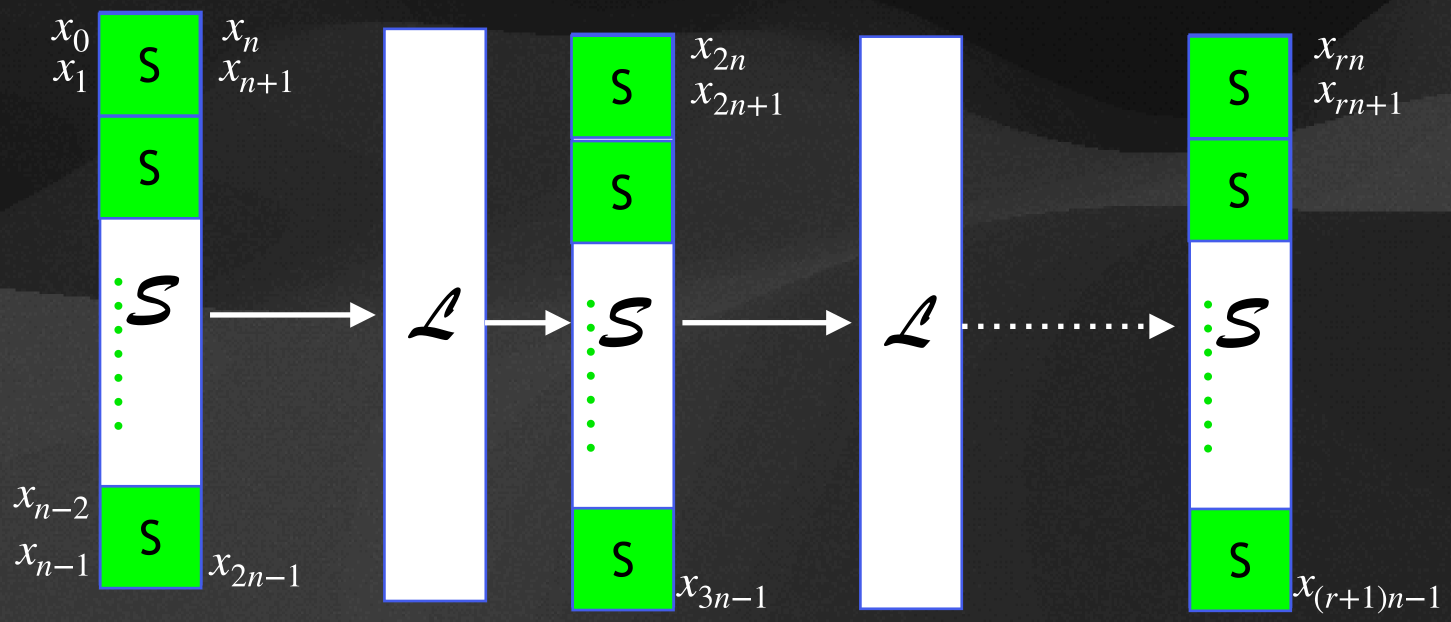




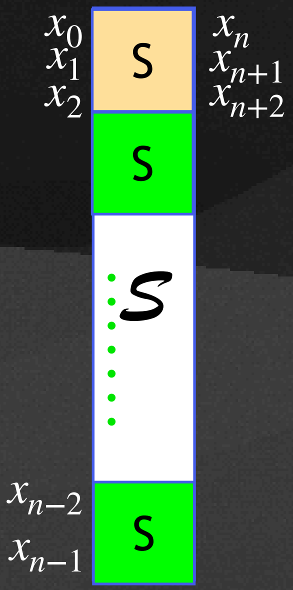


Finding trails using CRHS equations

Label the state bits



CRHS equation for DDT/LAT



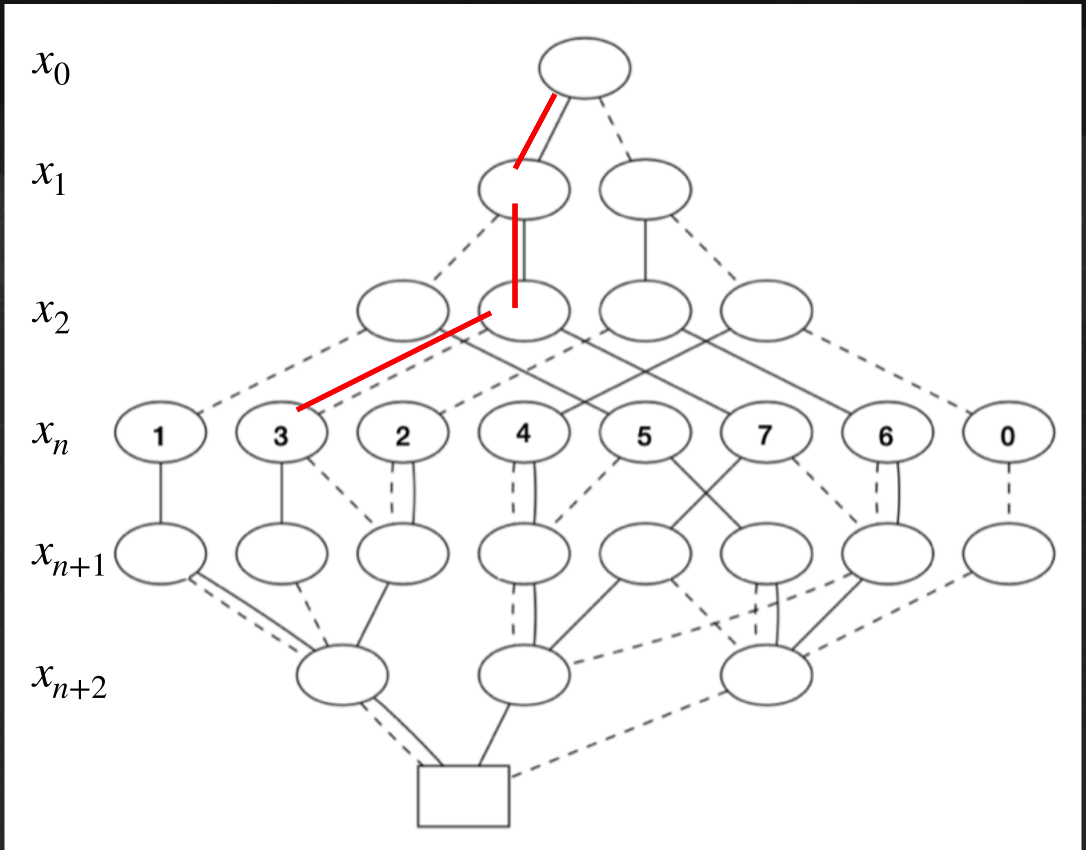
x	0	1	2	3	4	5	6	7
S[x]	0	1	3	6	7	4	5	2

S-box

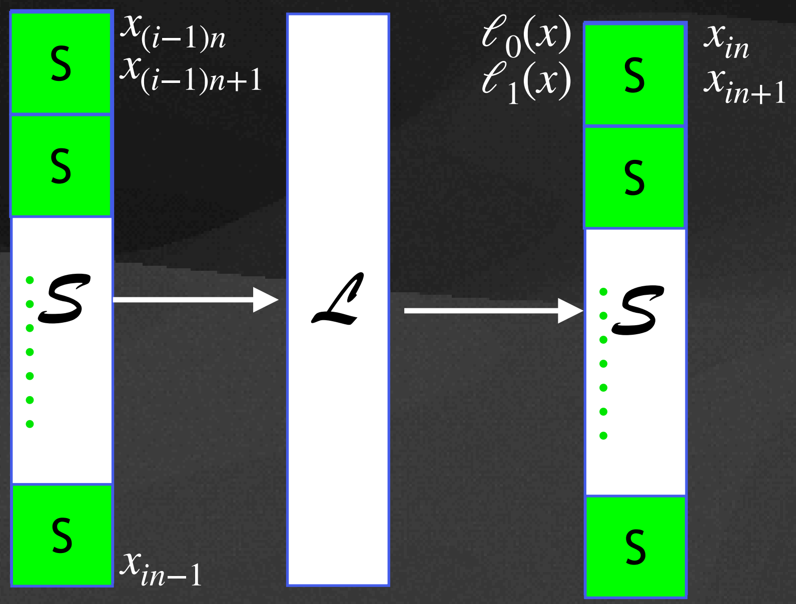
Output

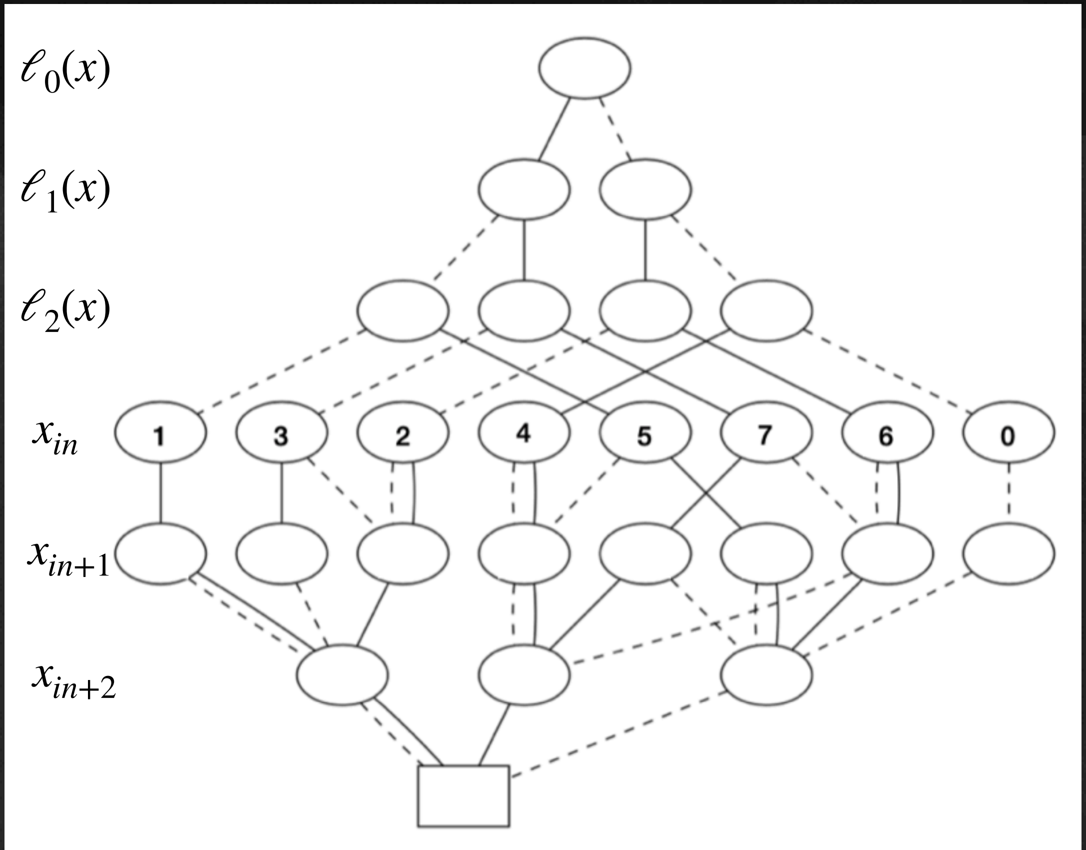
Input	0	1	2	3	4	5	6	7
0	8	0	0	0	0	0	0	0
1	0	2	0	2	0	2	0	2
2	0	0	2	2	0	0	2	2
3	0	2	2	0	0	2	2	0
4	0	0	0	0	2	2	2	2
5	0	2	0	2	2	0	2	0
6	0	0	2	2	2	2	0	0
7	0	2	2	0	2	0	0	2

DDT

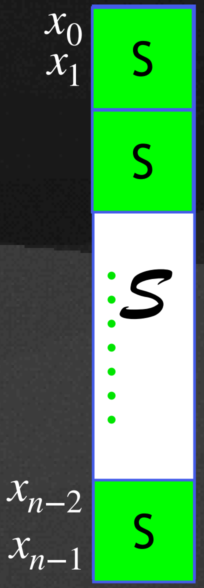


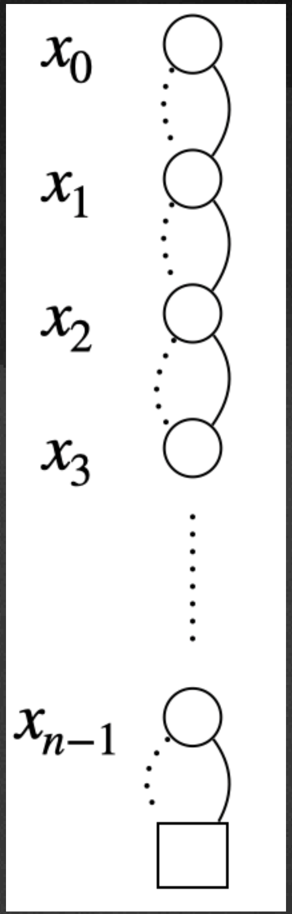
CRHS equation for DDT/LAT





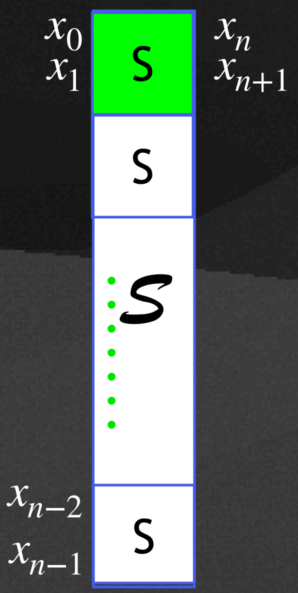
Initial master CRHS equation

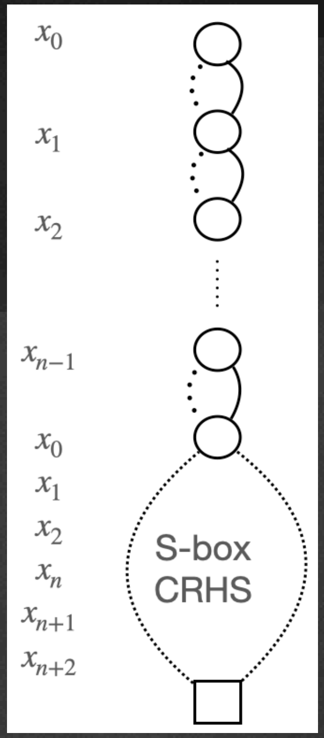




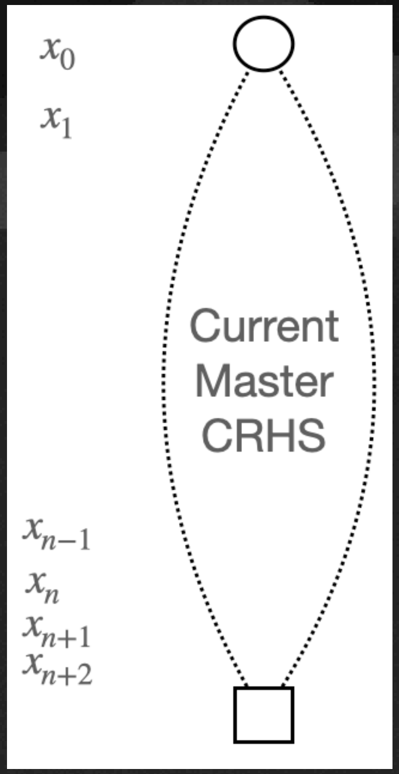
Initial Master CRHS equation has $n+1$ nodes and contains all 2^n possible inputs to $\mathcal{S}$ in first round

First join

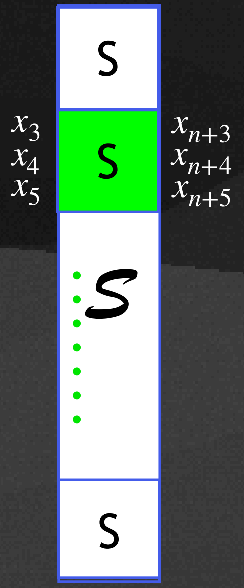


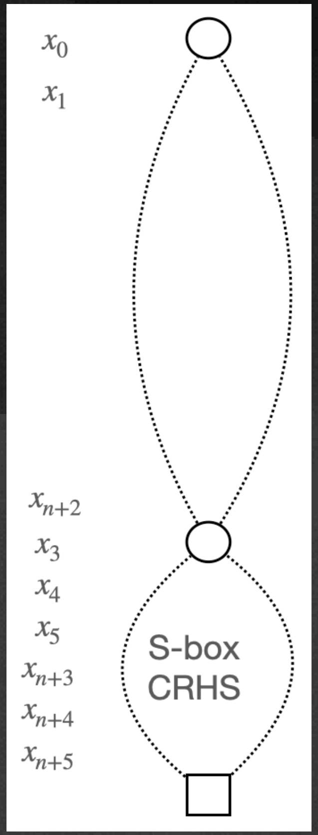


Absorb
dependencies

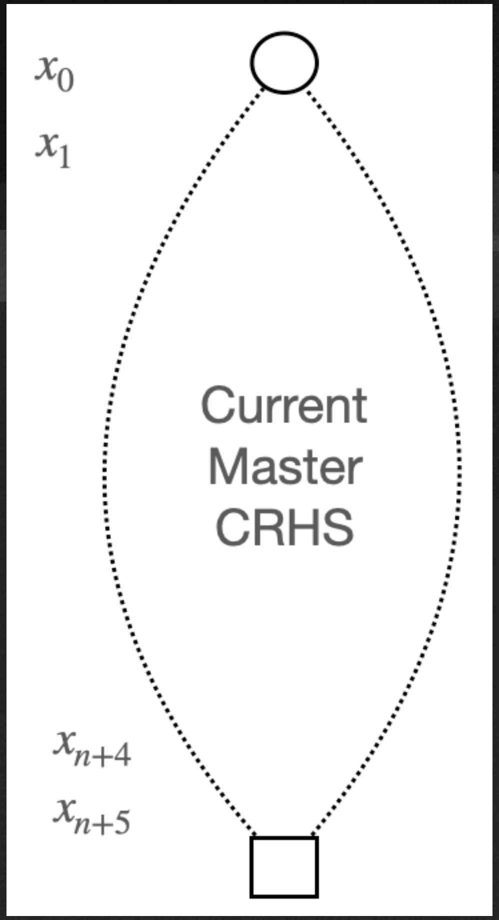


Second join

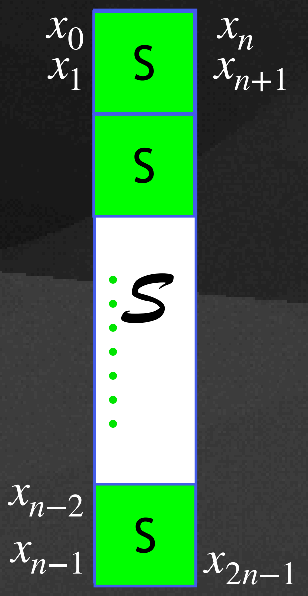


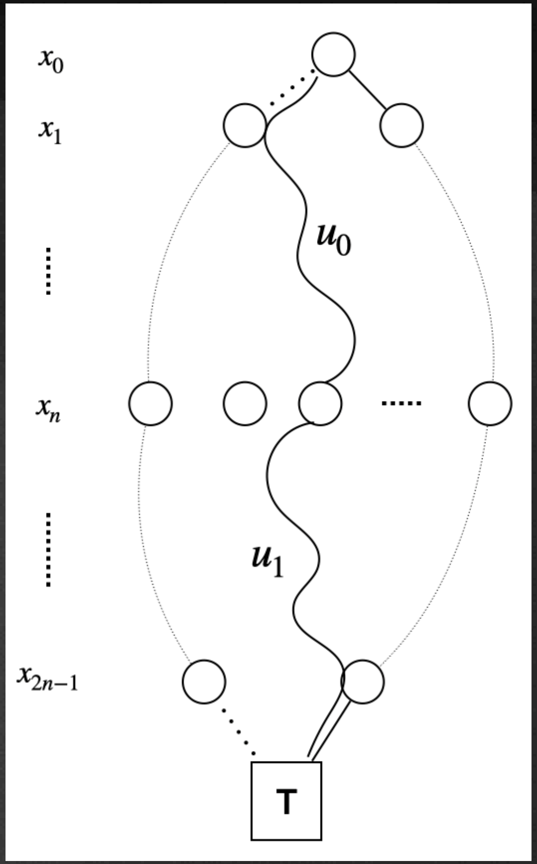


Absorb dependencies



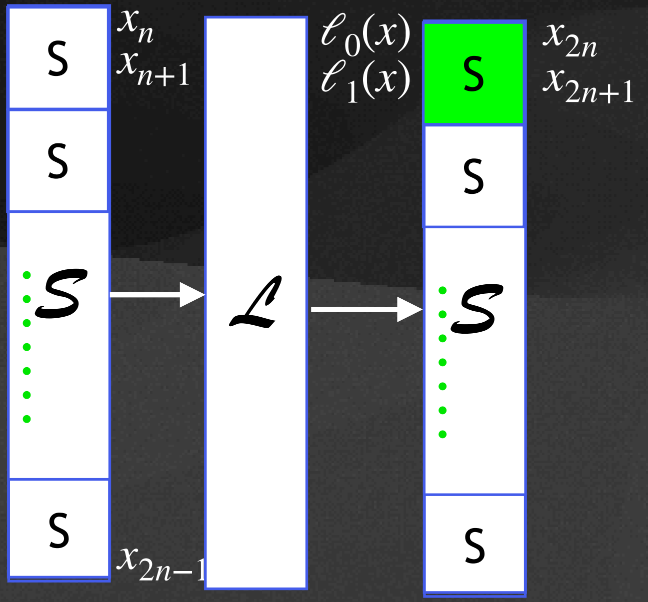
Master CRHS after first round

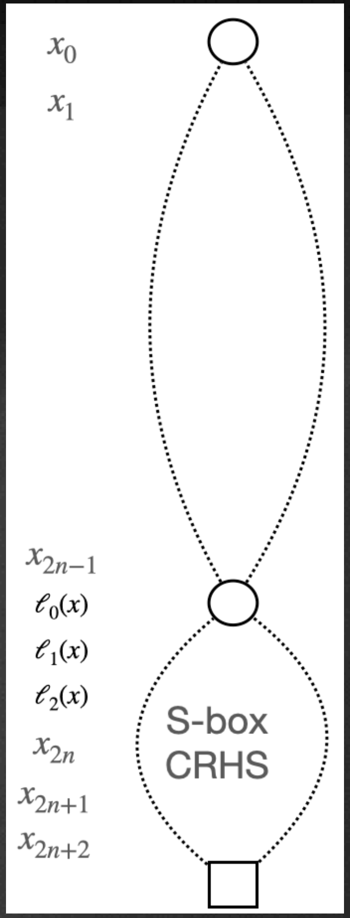




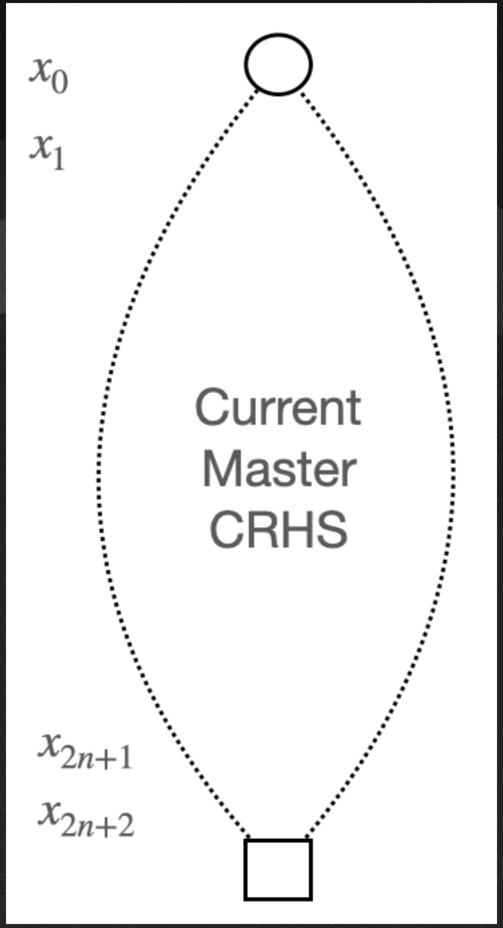
CRHS contains starts of all possible trails $(u_0, u_1, \ldots)$

Second round

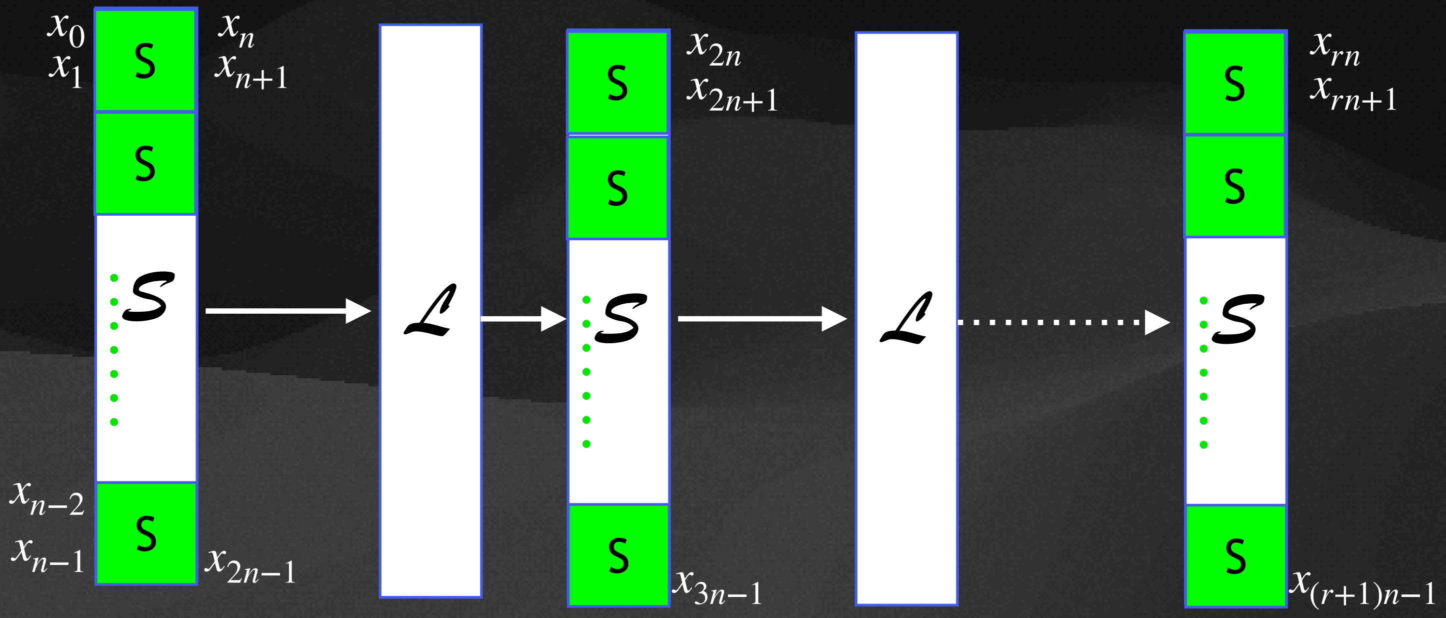


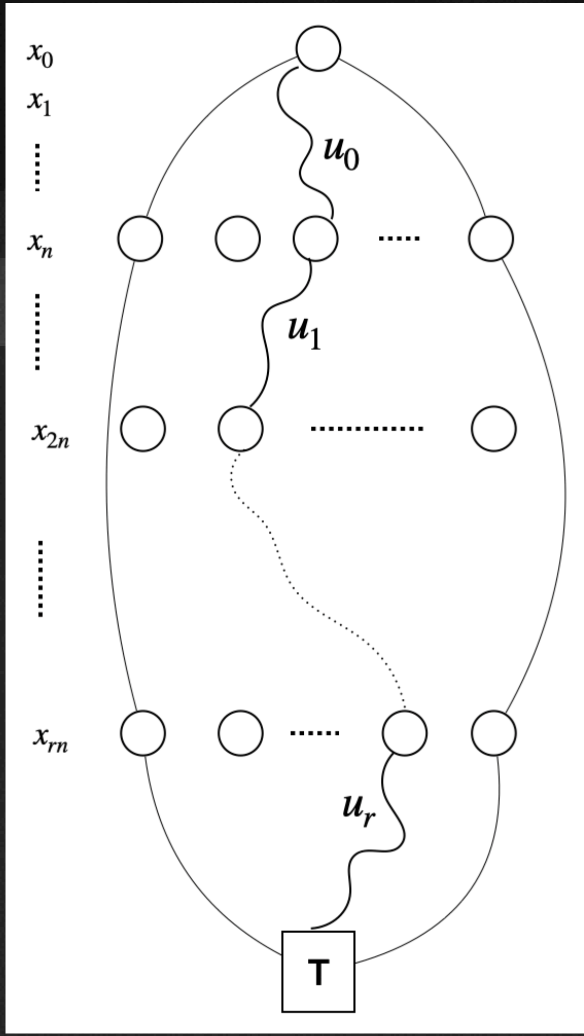


Absorb dependencies



After last join+absorb



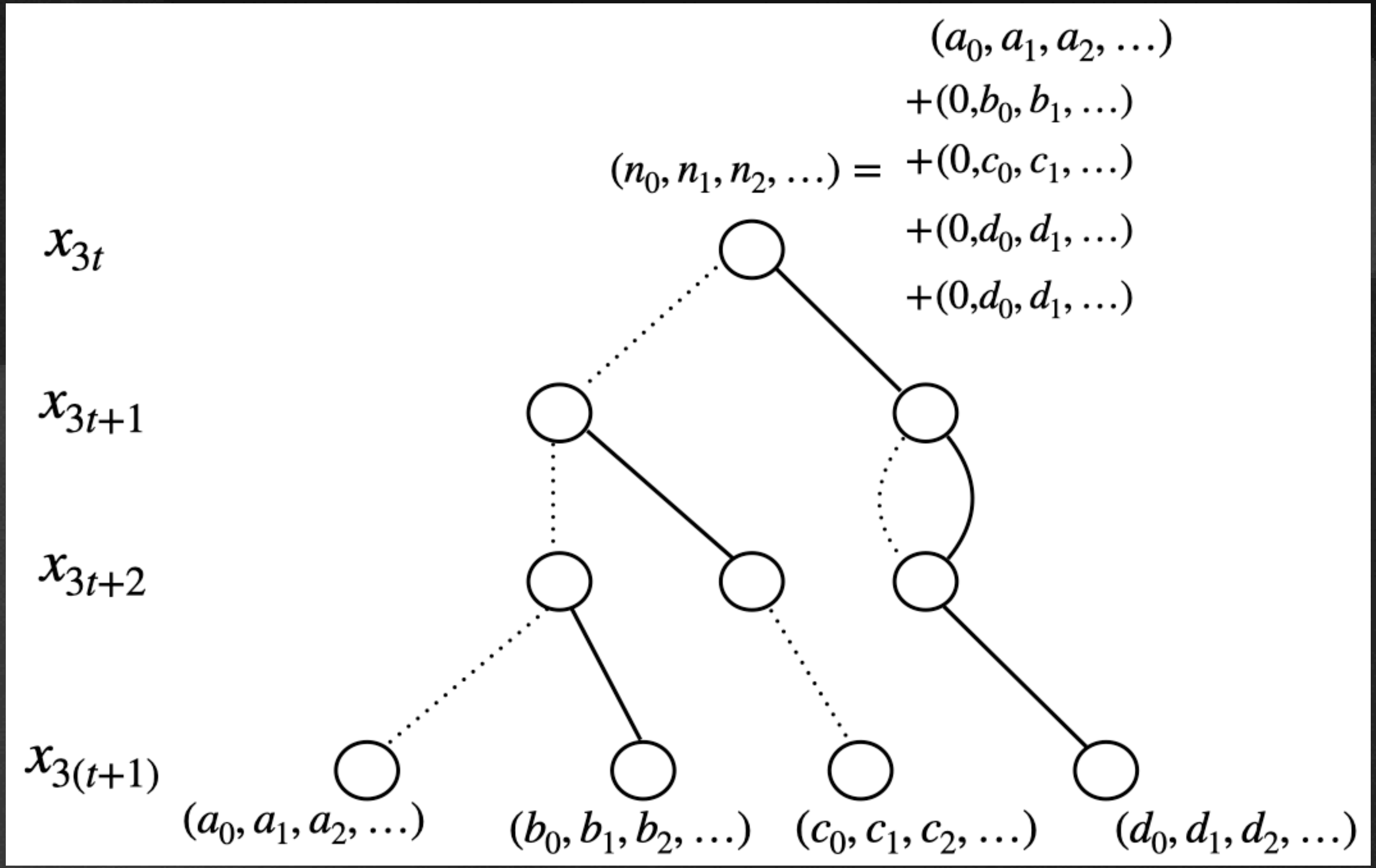


Paths in master CRHS equation encodes all possible trails in cipher

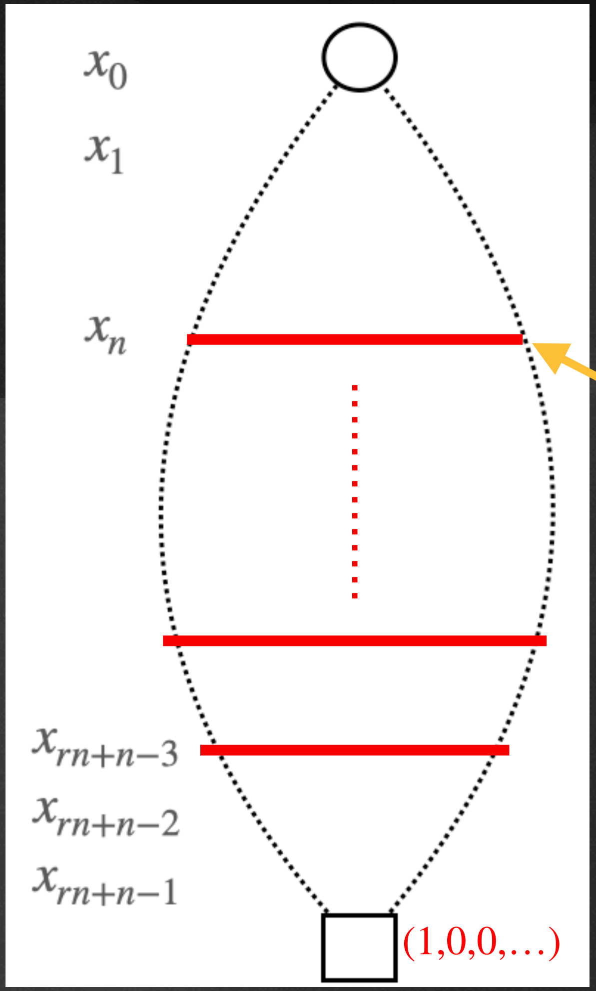
Counting active S-boxes

- Can count number of trails with i active S-boxes, $0 \leq i \leq rm$
- Linear complexity (in the number of nodes)
- Associate vector $(n_0, n_1, \ldots, n_{rm}) \in \mathbb{Z}^{rm+1}$ with each node
- n_i indicates number of sub-trails below node with i active S-boxes

Counting active S-boxes



Counting active S-boxes



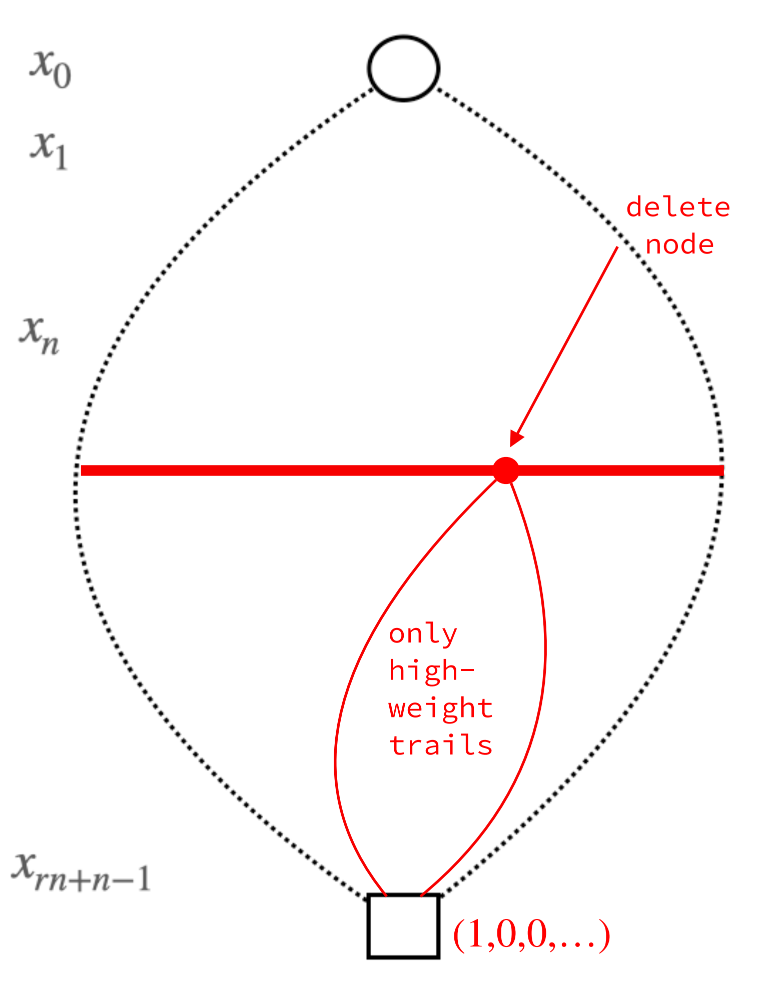
$(n_0, n_1, n_2, \ldots)$-vectors on this level indicate how many trails there are with exactly i active S-boxes

Pruning

- Joining and absorbing makes number of nodes, $\mathcal{N}$, in Master CRHS equation grow
- Worst case: one absorb doubles number of nodes
- If hardware can handle CRHS equation with up to μ nodes, let $\sigma = \mu/2^t$ be the limit for pruning (t-bit S-box)
- Delete nodes when $\mathcal{N} > \sigma$
- Guarantee: after next join and absorb of b dependencies $\mathcal{N} < \mu$

Pruning strategy

- Delete nodes from level with most nodes (widest level)
- Compute number of active S-boxes in sub-trails below widest level
- Delete nodes with only high-weight sub-trails below itself

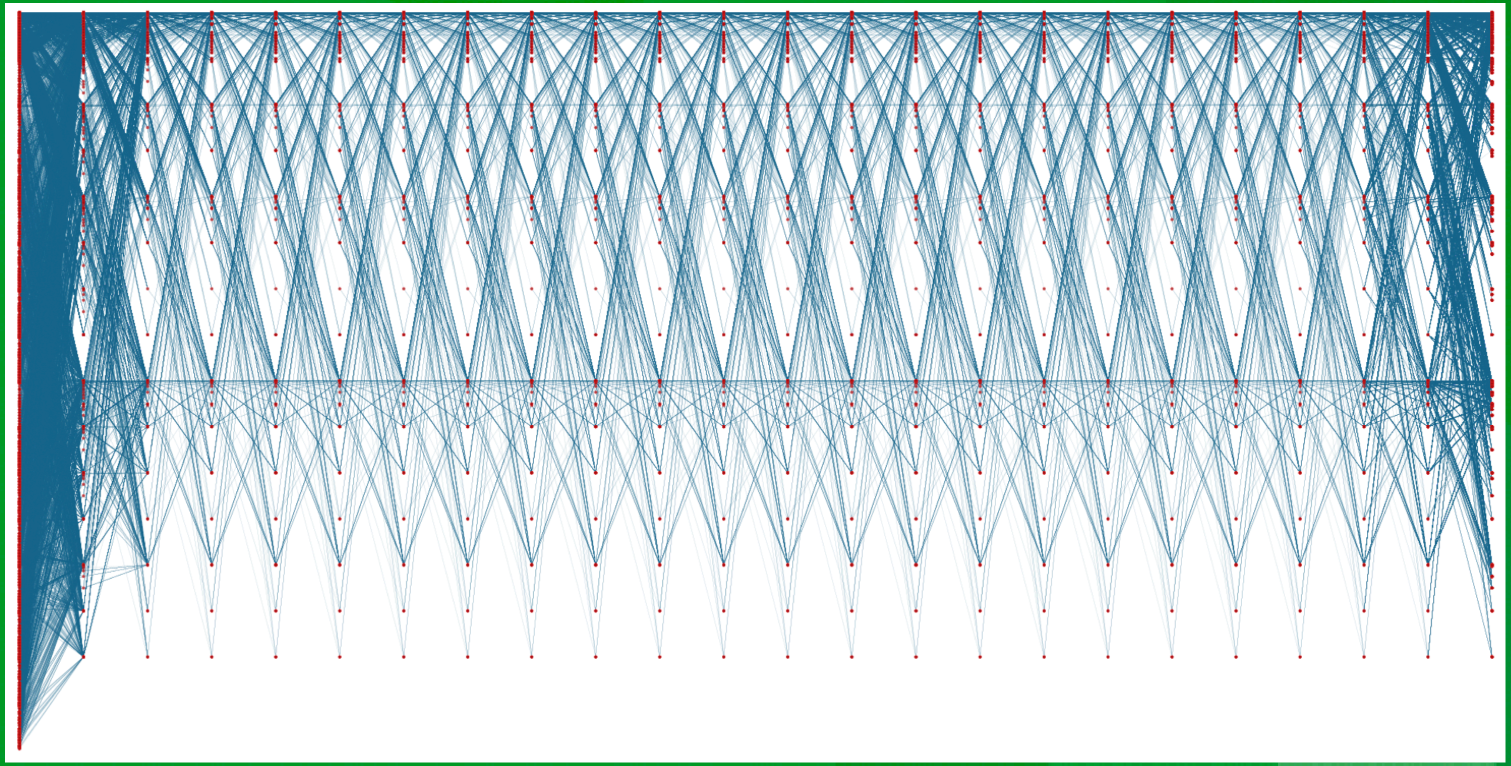


Pathfinder and CryptaGraph

Software tools

- Method using CRHS equations made into software tool called Pathfinder
- CryptaGraph tool implementing method in [1]
- Only requires reference implementation (in Rust) of cipher to use, no need to understand underlying methods

CryptaGraph method



Every node represents one n-bit state u_i

Nodes one same level are all n-bit states considered in given round

Edges are all valid transitions from one round to next

Comparison of methods

CryptaGraph	Pathfinder
Cipher state represented by single node	Cipher state represented by partial path
States to include in search must be determined beforehand	States in search emerge dynamically at runtime
Computing weight of hull in aggregate fashion, works for exponentially large hulls	Computing weight of hull must be done one path at a time, does not work on exponentially large hulls

Strong advantages

Combining CryptaGraph and Pathfinder?

- Combining the tools should make strongest trail-search algorithm
- High-level idea:
  1. Run Pathfinder to find states that actually occur in low-weight trails
  2. Run CryptaGraph with nodes representing these states

Linear trail results

Cipher (Total Rounds, block size)	Rounds	Soft Lim	Hull Size (Used, Found)	ELP	CG result
MIDORI64 (16, 64)	6	2^{18}	$2^{21.62}$, $2^{23.89}$	$2^{-85.03}$	$2^{-53.02}$
	7	2^{18}	2^{26}, $2^{29.66}$	$2^{-108.42}$	$2^{-62.88}$
PRESENT (31, 64)	23	2^{18}	2^{26}, $2^{37.03}$	$2^{-69.23}$	$2^{-61.00}$
	24	2^{18}	2^{26}, $2^{38.60}$	$2^{-73.23}$	$2^{-63.61}$
	25	2^{18}	2^{26}, $2^{39.65}$	$2^{-76.54}$	$2^{-66.21}$
PRIDE (20, 64)	15	2^{18}	1, 1	$2^{-58.00}$	$2^{-58.00}$
	16	2^{18}	7, 7	$2^{-65.99}$	$2^{-63.99}$
PRINCE ($2 \cdot 6$, 64)	$2 \cdot 3$	2^{18}	19, 19	$2^{-55.57}$	$2^{-54.00}$
	$2 \cdot 4$	2^{18}	214, 214	$2^{-92.90}$	$2^{-63.82}$
PUFFIN (32, 64)	32	2^{18}	2^{26}, $2^{52.55}$	$2^{-83.69}$	$2^{-51.90}$
QARMA ($2 \cdot 8$, 64)	$2 \cdot 3$	2^{18}	612, 1433	$2^{-95.75}$	$2^{-53.71}$
RECTANGLE (25, 64)	12	2^{18}	$2^{16.66}$, $2^{16.66}$	$2^{-56.75}$	$2^{-52.27}$
	13	2^{18}	$2^{17.16}$, $2^{17.16}$	$2^{-64.22}$	$2^{-58.14}$
	14	2^{18}	$2^{16.51}$, $2^{16.51}$	$2^{-68.48}$	$2^{-62.98}$

Differential trail results

Cipher (Total Rounds, block size)	Rounds	Soft Lim	Hull Size (Used, Found)	EDP	CG result
KLEIN	5	2^{18}	8, 8	$2^{-44.39}$	$2^{-45.91}$
(12, 64)	6	2^{22}	4, 4	$2^{-55.25}$	$2^{-69.00}$
LED (32, 64)	4	2^{22}	6, 18	$2^{-55.61}$	$2^{-49.42}$
MANTIS_7 $(2 \cdot 8, 64)$	$2 \cdot 4$	2^{22}	$2^{24.94}$, $2^{26.64}$	$2^{-100.87}$	$2^{-47.98}$
MIDORI64	6	2^{22}	$2^{20.28}$, $2^{21.50}$	$2^{-63.60}$	$2^{-52.37}$
(16, 64)	7	2^{22}	$2^{23.82}$, $2^{25.49}$	$2^{-71.75}$	$2^{-61.22}$
PRESENT	15	2^{18}	$2^{15.42}$, $2^{15.42}$	$2^{-65.69}$	$2^{-58.00}$
(31, 64)	16	2^{18}	$2^{15.97}$, $2^{16.29}$	$2^{-69.71}$	$2^{-61.80}$
	17	2^{18}	$2^{17,76}$, $2^{17.76}$	$2^{-74.87}$	$2^{-63.52}$
PRIDE	15	2^{22}	1, 1	$2^{-58.00}$	$2^{-58.00}$
(20, 64)	16	2^{22}	1, 1	$2^{-64.00}$	$2^{-63.99}$
PRINCE	$2 \cdot 3$	2^{22}	16, 20	$2^{-49.45}$	$2^{-55.91}$
$(2 \cdot 6, 64)$	$2 \cdot 4$	2^{22}	36, 36	$2^{-80.67}$	$2^{-67.32}$
PUFFIN (32, 64)	32	2^{18}	2^{26}, $2^{37.25}$	$2^{-79.71}$	$2^{-59.63}$

Trails for Klein and Prince

Klein

MSB ... LSB	Active S-boxes
0000050000050000	
S-box Layer	2
0000020000020000	
Linear Layer	
0600040200000000	
S-box Layer	3
0100030500000000	
Linear Layer	
0909060001030201	
S-box Layer	7
080e040004040a0e	
Linear Layer	
080c000000000604	
S-box Layer	4
0b0d000000000809	
Linear Layer	
000000000d0a0000	
S-box Layer	2
0000000002060000	
Linear Layer	
04000e0e00000000	
S-box Layer	3
0100030300000000	

Prince

MSB ... LSB	Active S-boxes
0000000000000101	
S-box Layer	2
0000000000000808	
Linear Layer	
0008000008000000	
S-box Layer	2
0008000004000000	
Linear Layer	
8040040840800000	
S-box Layer	6
8080040450500000	
Middle involution	
8080040450500000	
S-box Layer	6
8040040840800000	
Linear Layer	
0008000004000000	
S-box Layer	2
0008000008000000	
Linear Layer	
0000000000000808	
S-box Layer	2
0000000000000101	

12-round Prince trail

- Designers of Prince prove that a 4-round trail in Prince must contain at least 16 active S-boxes
- Conclude that trails in full 12-round Prince must have at least 48 active S-boxes
- Pathfinder finds trail with exactly 48 active S-boxes when run on 12-round Prince

Trail is non-iterative with number of active S-boxes in each round
2,6,6,2,2,6,6,2,2,6,6,2