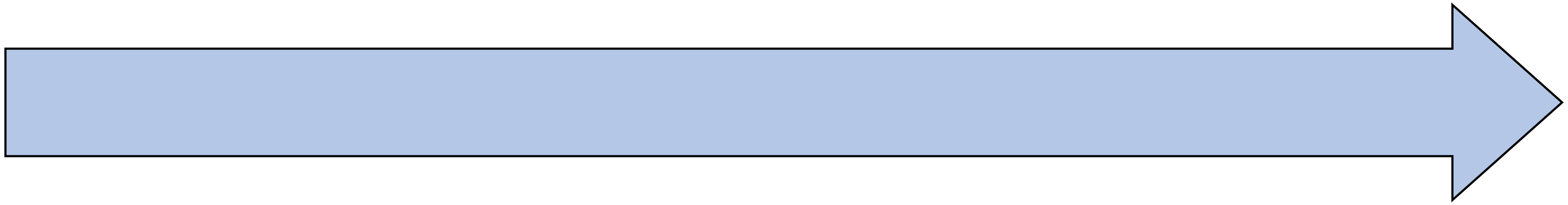
Finding a Nash Equilibrium is No Easier than Breaking Fiat-Shamir

Arka Rai Choudhuri   Pavel Hubáček   Chethan Kamath
Krzysztof Pietrzak   Alon Rosen   Guy Rothblum

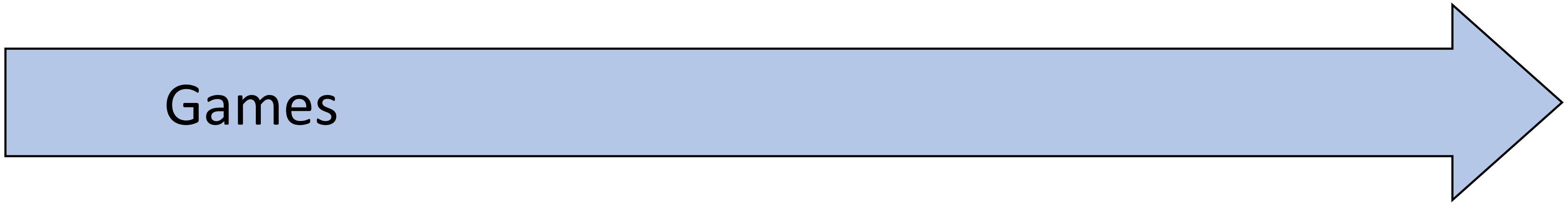
JHU Theory Seminar

What is the Cryptographic Hardness in PPAD?

The Story Line



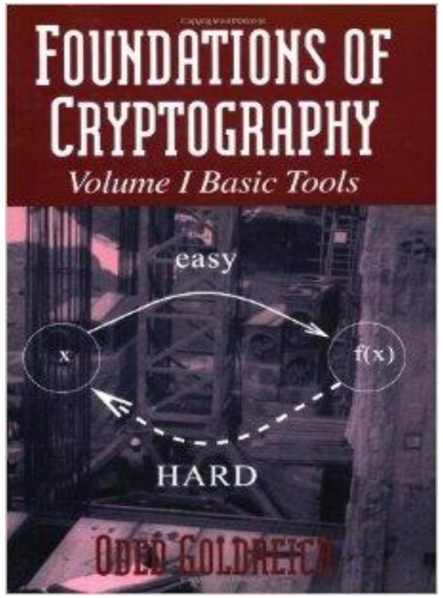
The Story Line



The Story Line

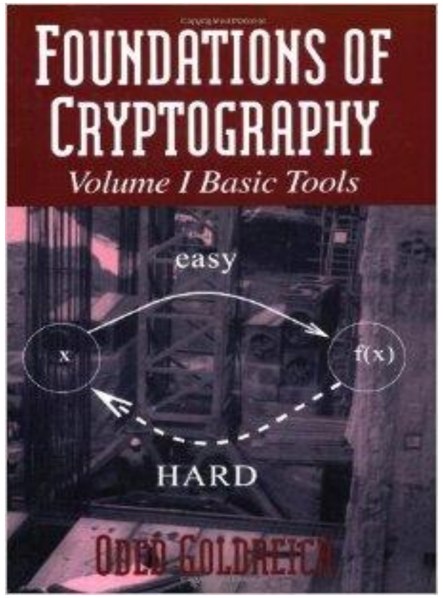
Games Complexity

The Story Line



Games Complexity Crypto

The Story Line



Games Complexity Crypto

The Story Line

Games Complexity Crypto

Game Theory and Nash Equilibrium

Game Theory and Nash Equilibrium

Game Theory and Nash Equilibrium

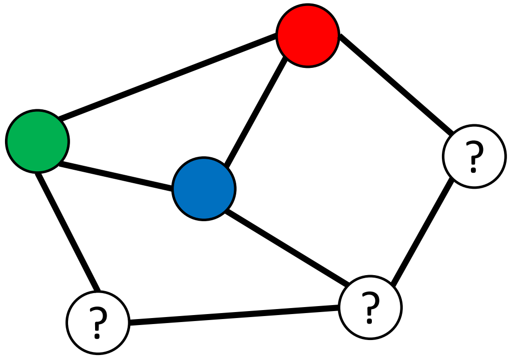
	Left	Right
Left	1 \ -1	-1\ 1
Right	-1 \ 1	1 \ -1

Game Theory and Nash Equilibrium

	Left	Right
Left	1 \ -1	-1\ 1
Right	-1 \ 1	1 \ -1

[Nash'51]: A (mixed) equilibrium always exists

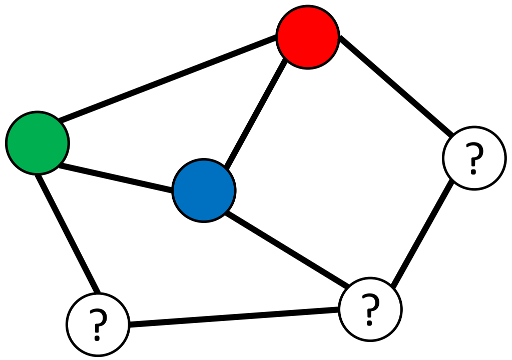
How hard is finding a Nash Equilibrium?



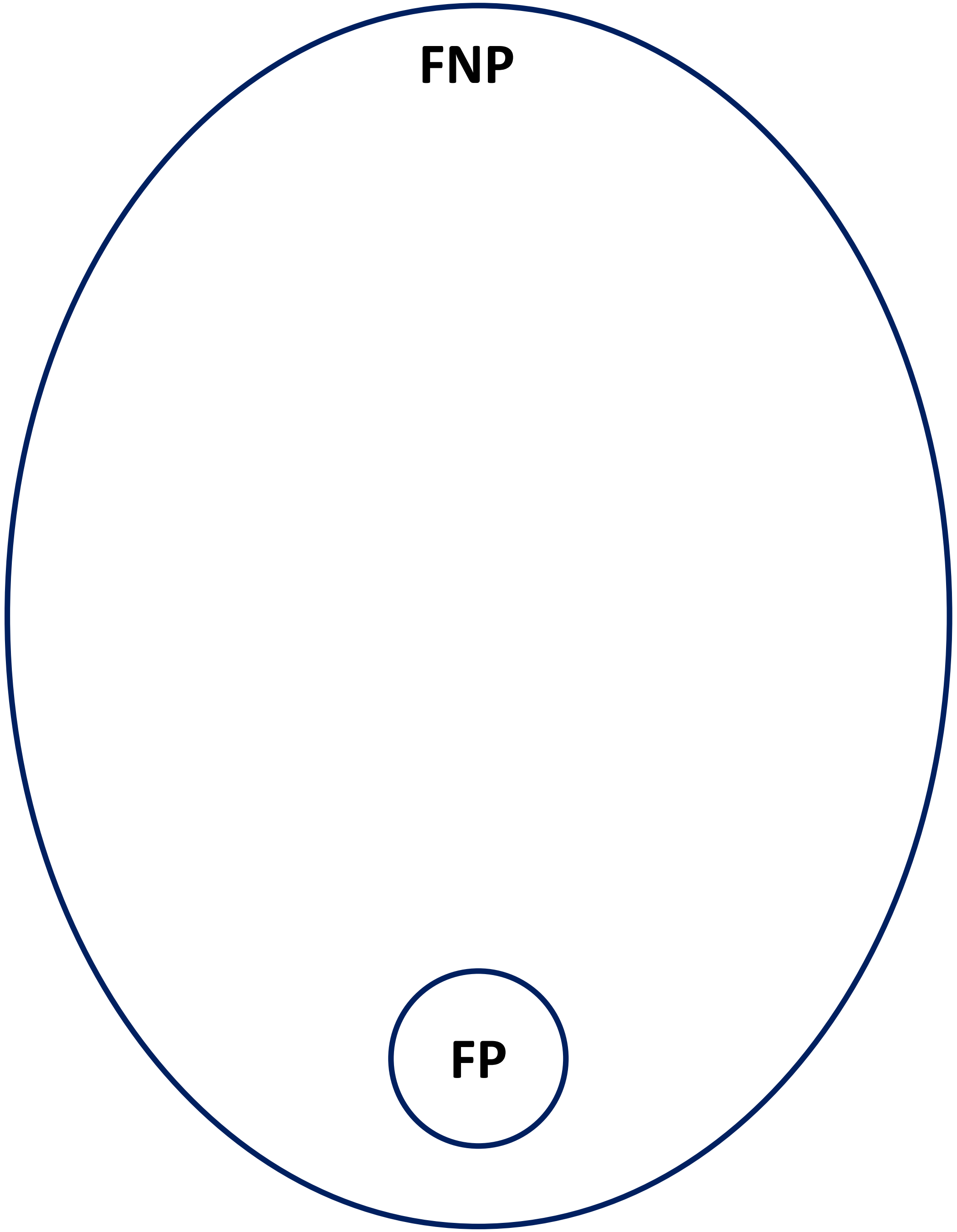
Reduction?

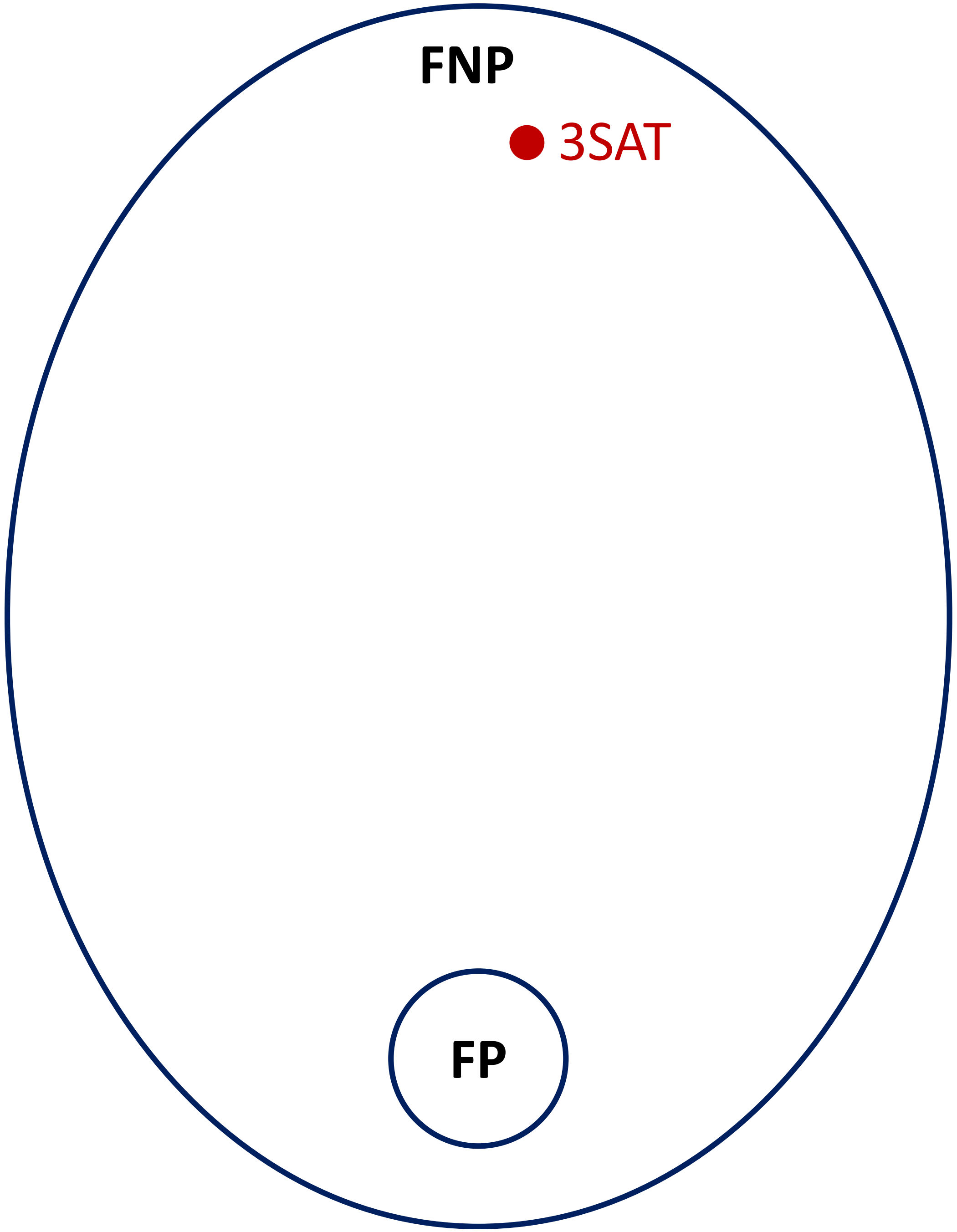
	Left	Right
Left	1 \ -1	-1\ 1
Right	-1 \ 1	1 \ -1

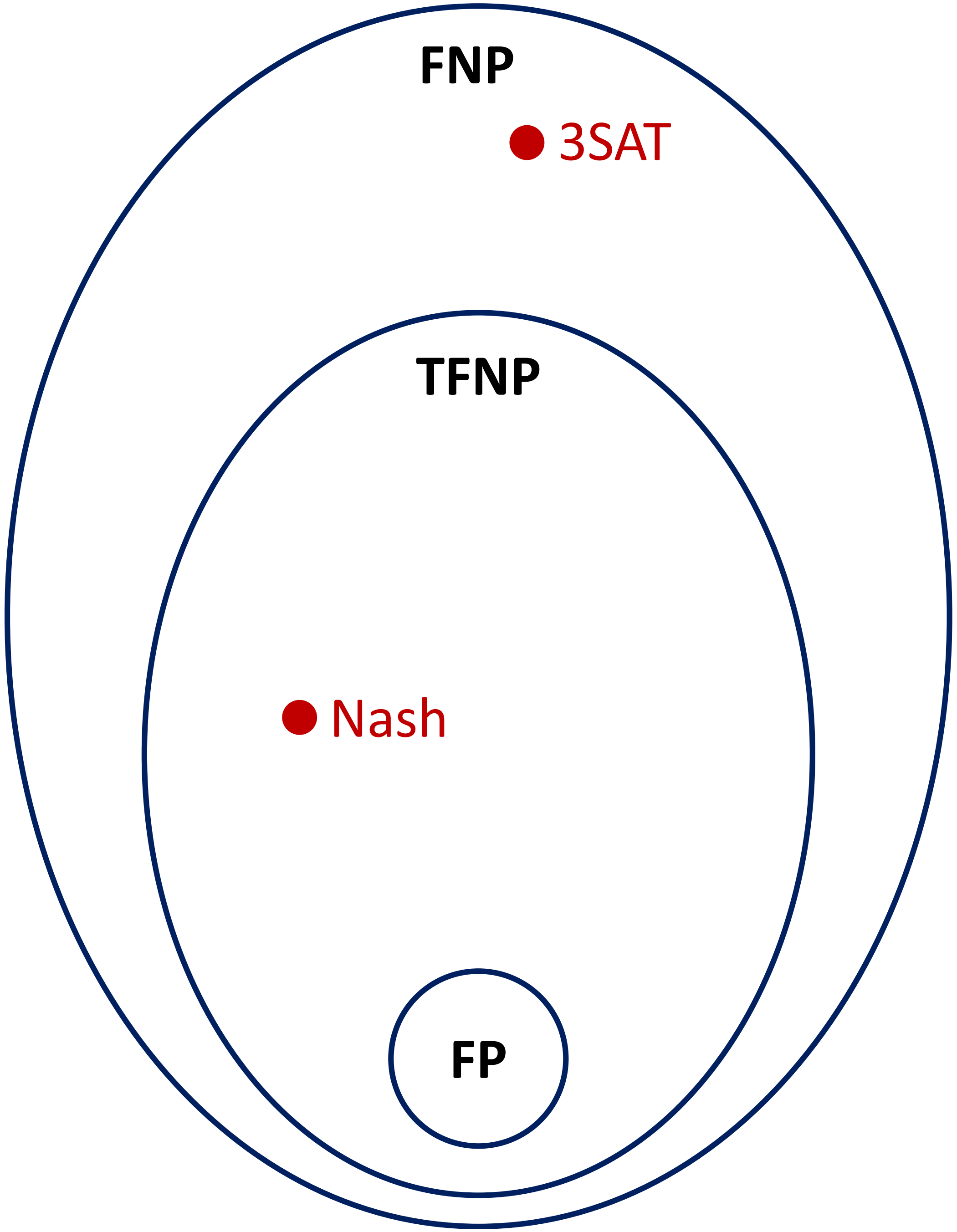
How hard is finding a Nash Equilibrium?

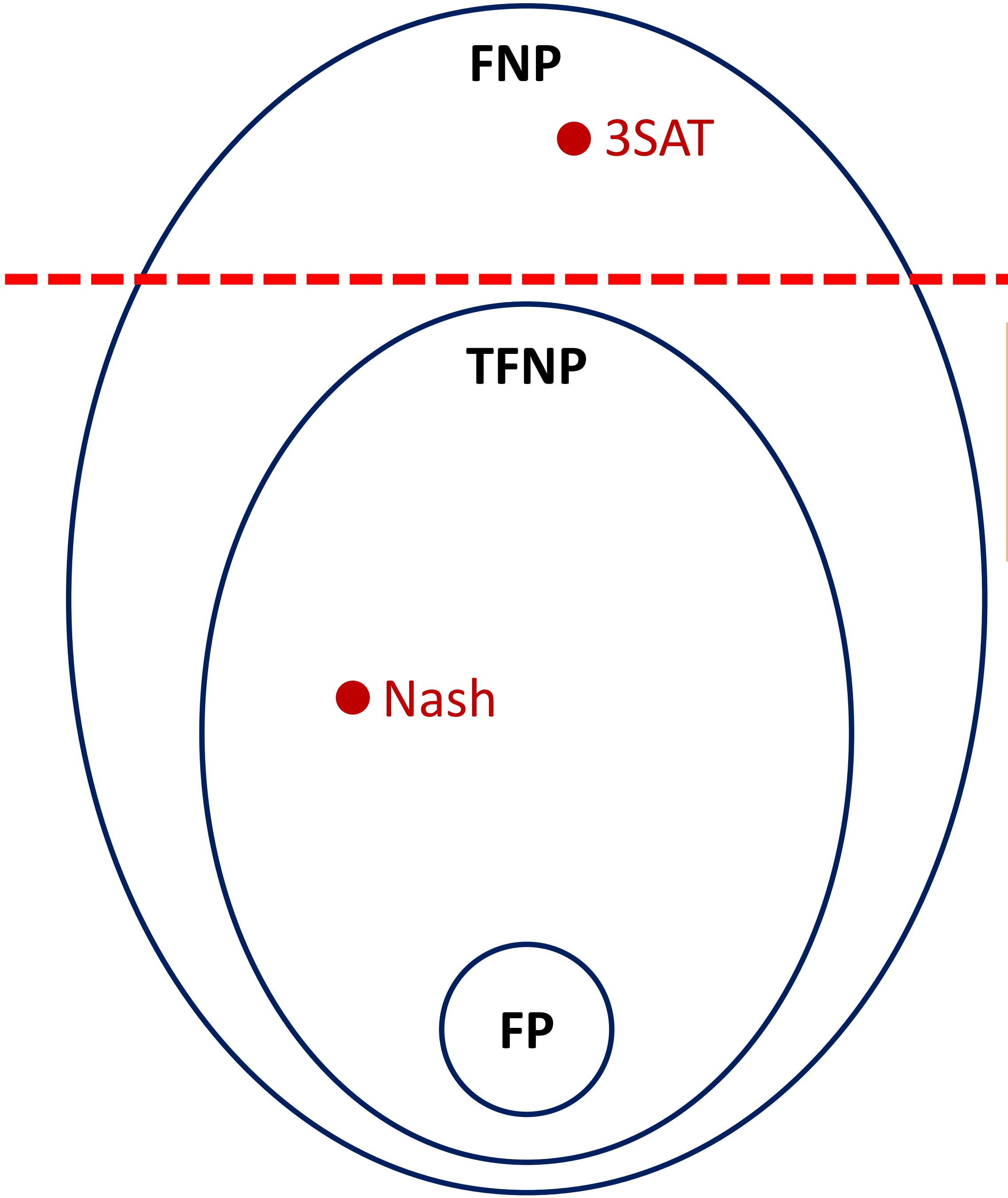


	Left	Right
Left	1 \ -1	-1\ 1
Right	-1 \ 1	1 \ -1


FNP
FP

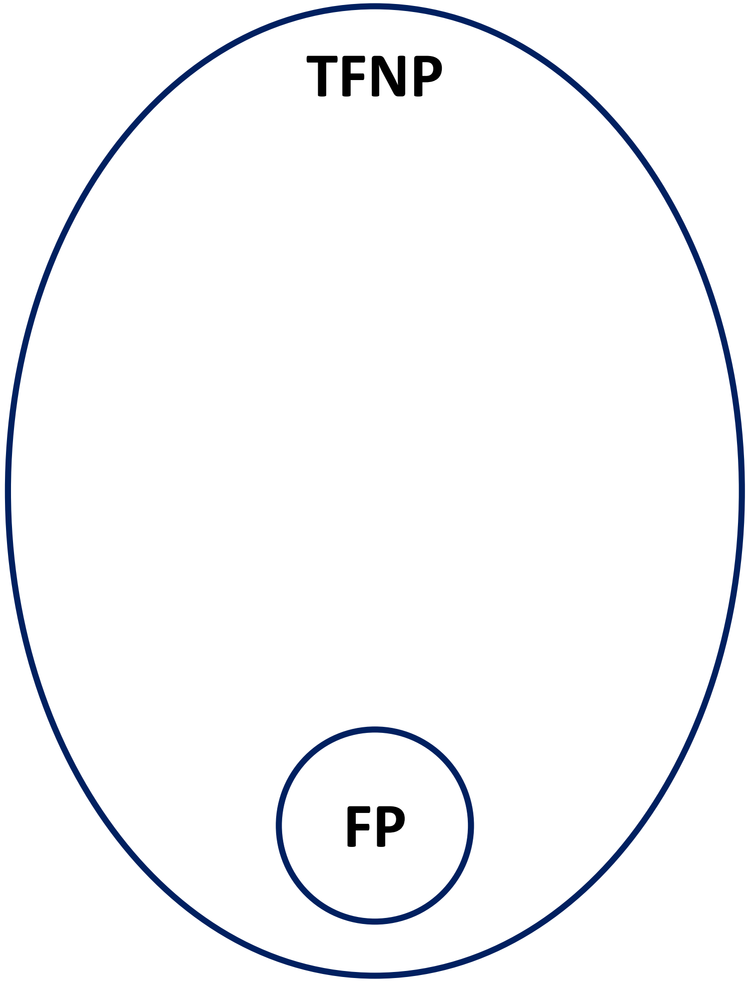

FNP
3SAT
FP


FNP
3SAT
TFNP
Nash
FP

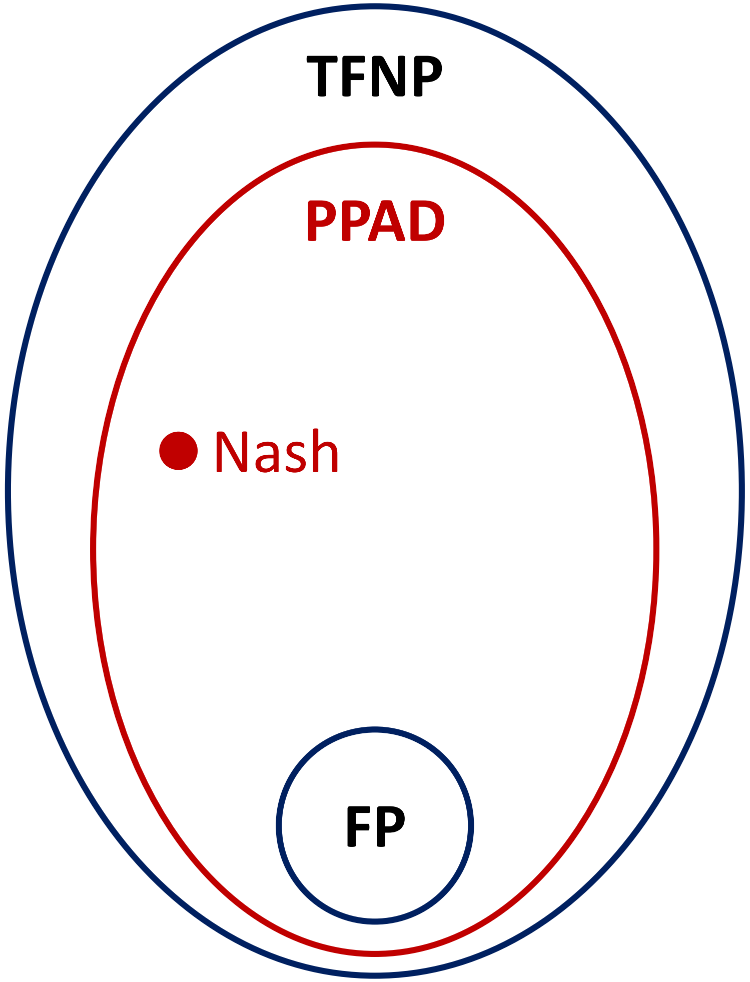


Not NP-hard unless NP=coNP
[Megiddo-Papadimitriou’89]

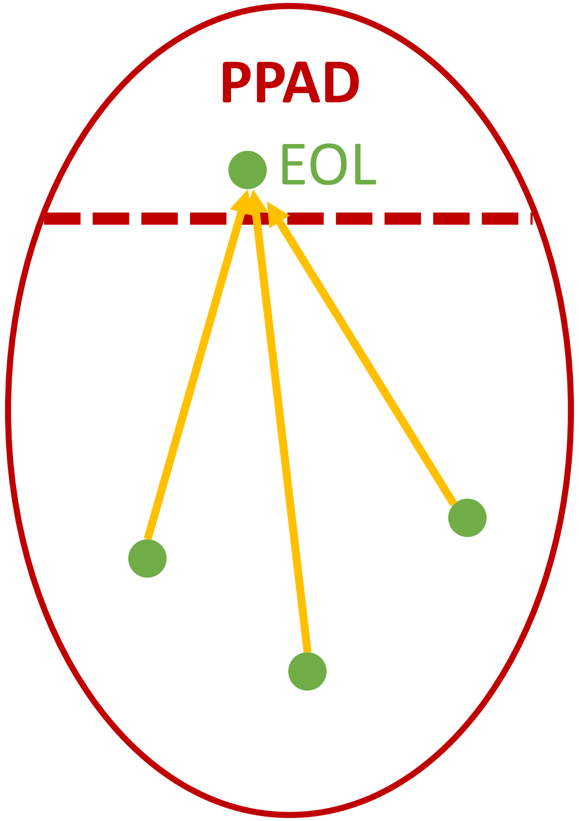
The Class PPAD [Papadimitriou'94]



The Class PPAD [Papadimitriou'94]

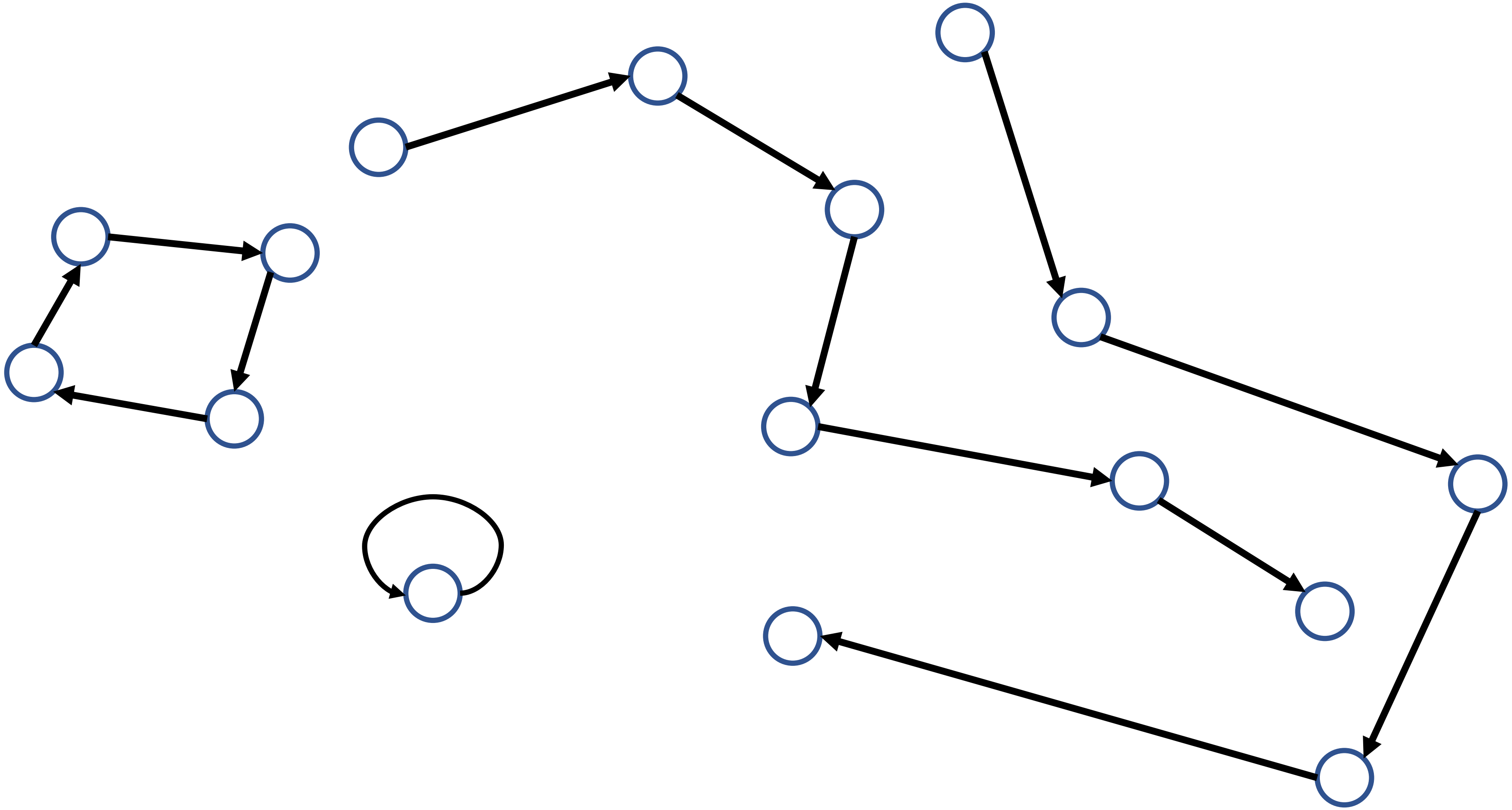


The Class PPAD [Papadimitriou'94]



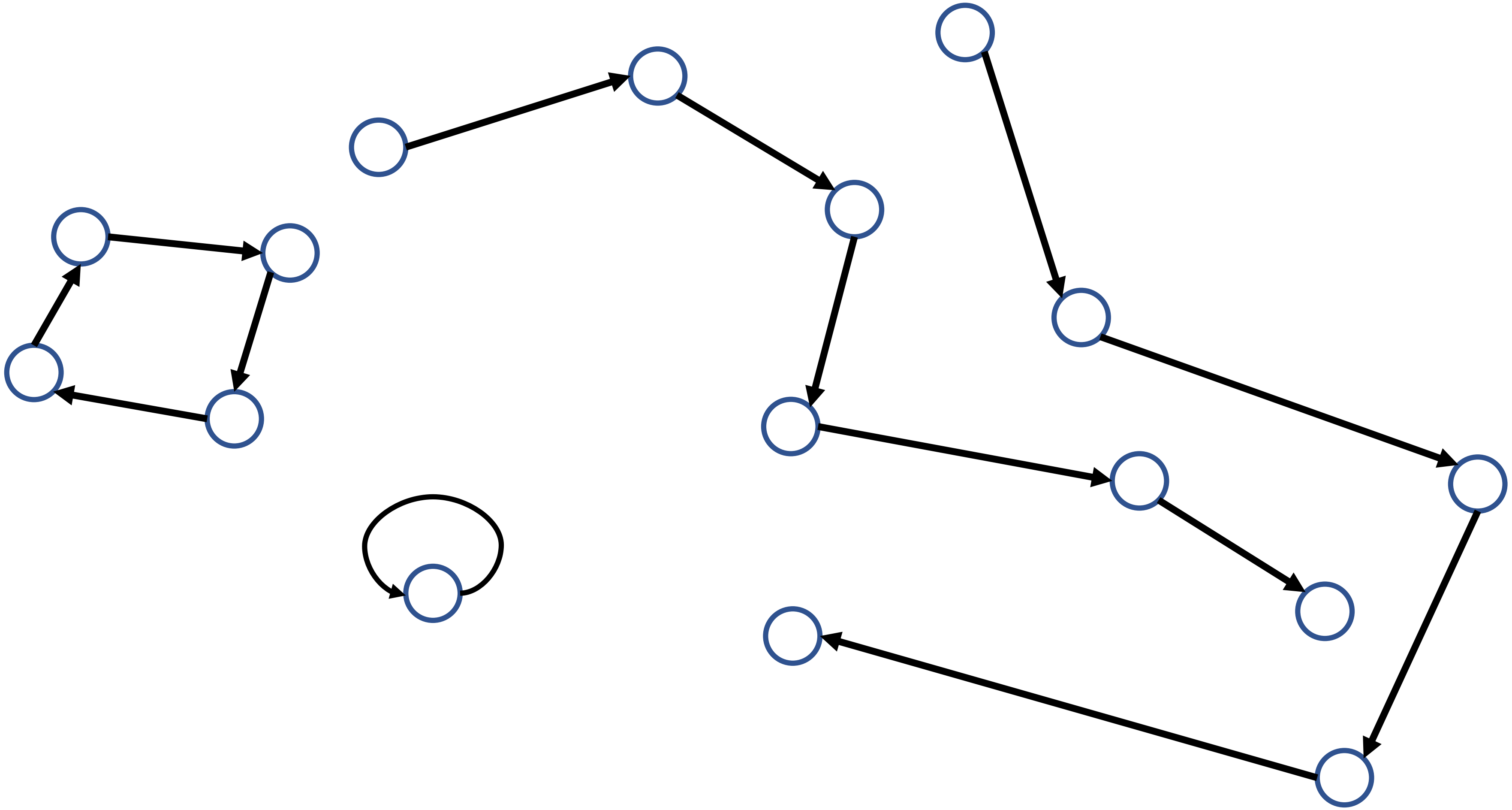
Defined through its complete problem:
END OF THE LINE (EOL)

End of the Line (EOL)

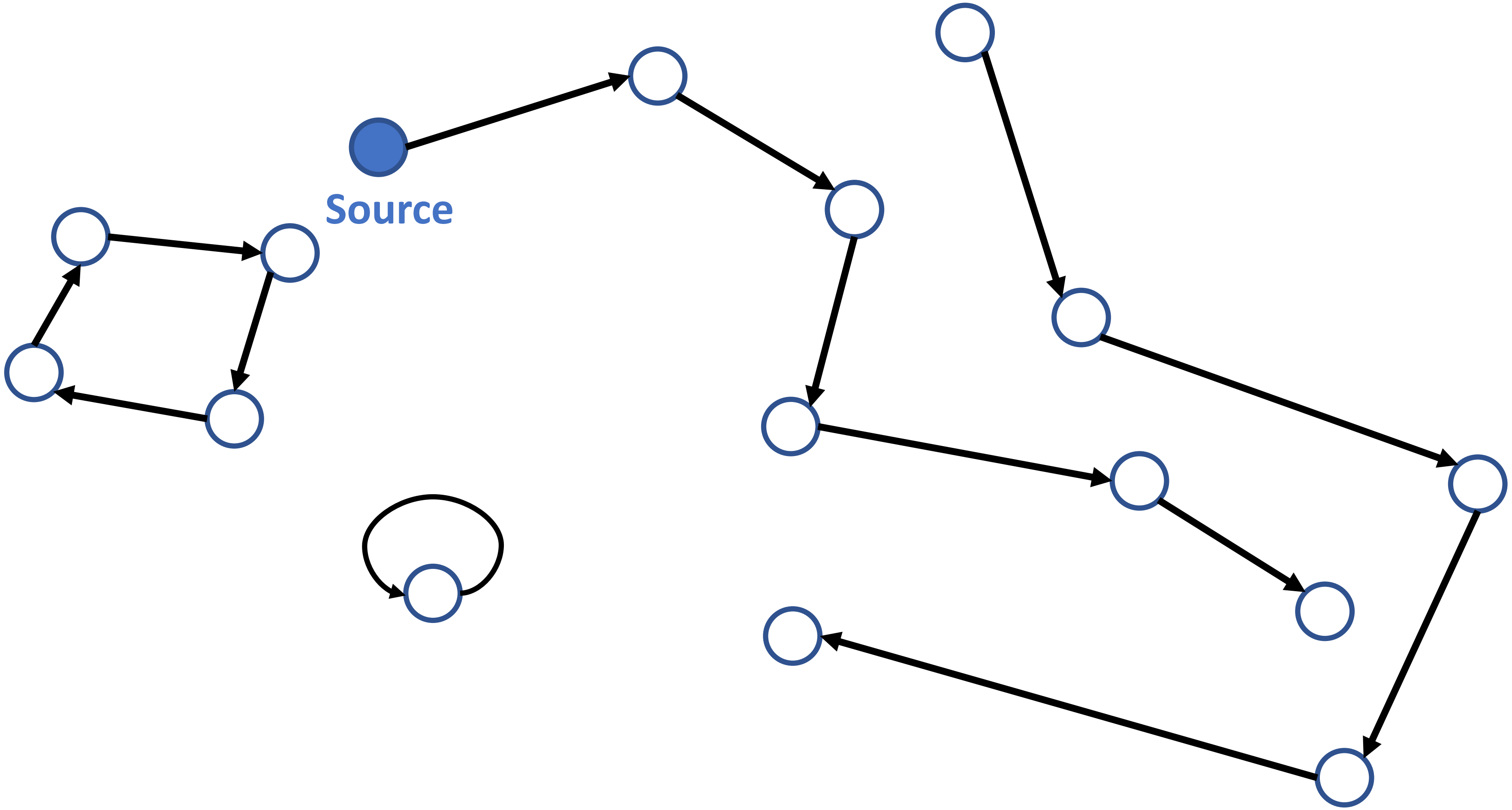


End of the Line (EOL)

Directed graph



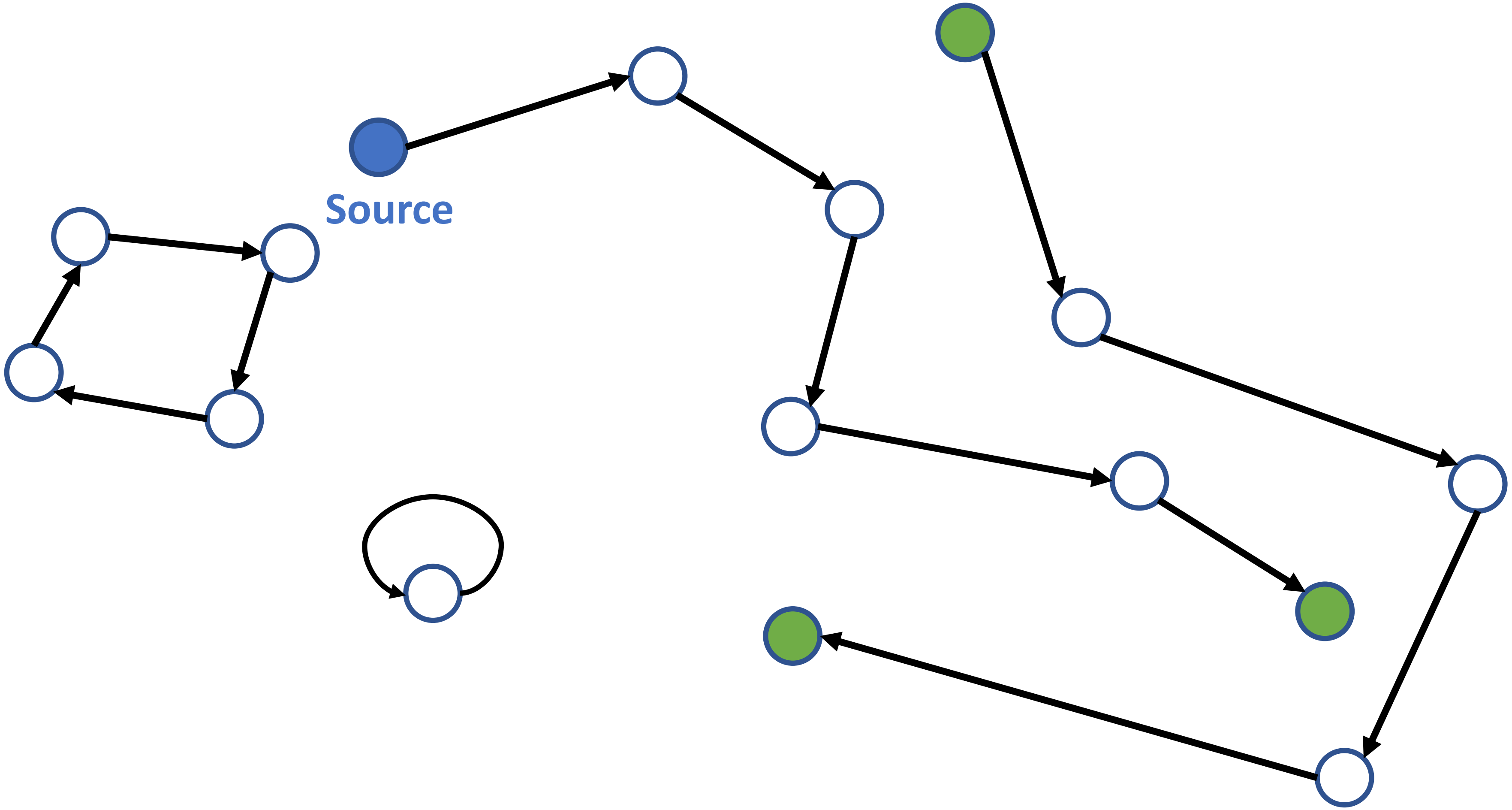
End of the Line (EOL)



Directed graph

Every node has in and out degree ≤ 1

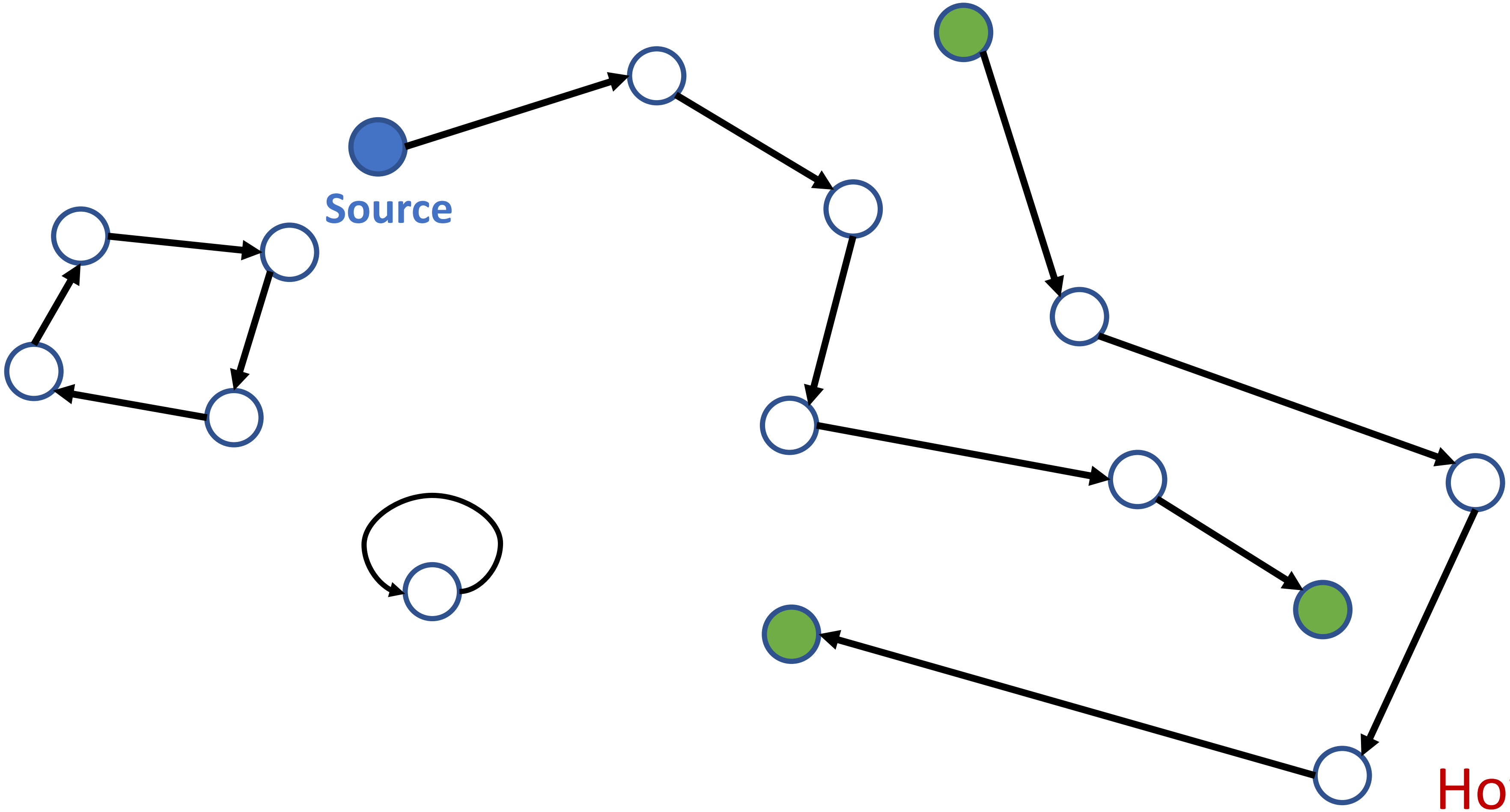
End of the Line (EOL)



Directed graph

Every node has in and out degree ≤ 1

End of the Line (EOL)

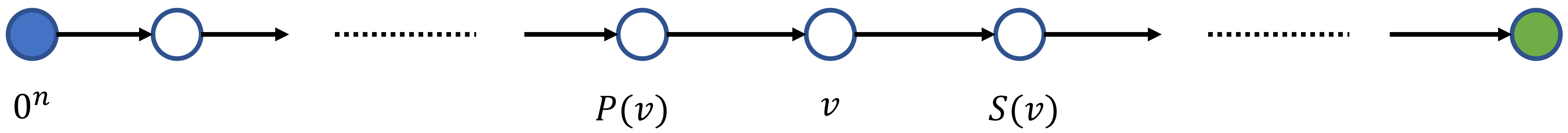


Directed graph

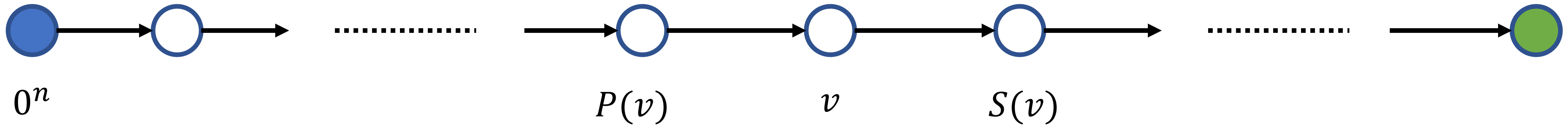
Every node has in and out degree ≤ 1

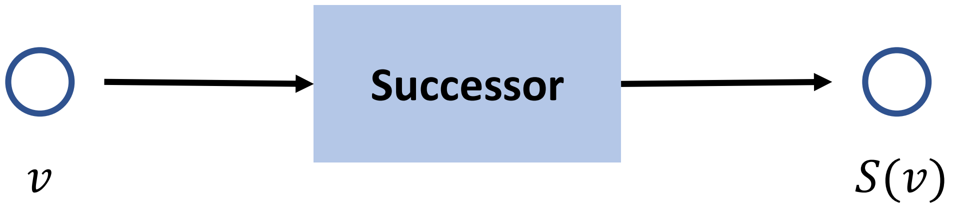
How easy is it to solve this?

End of the Line (EOL)

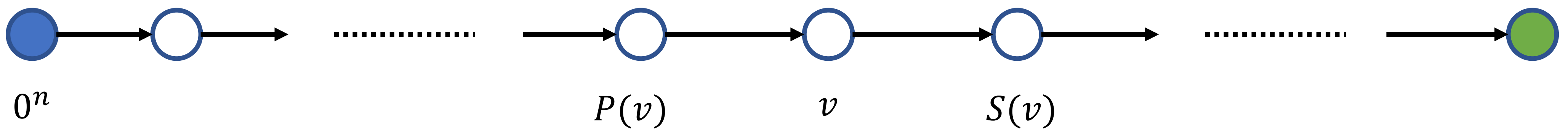


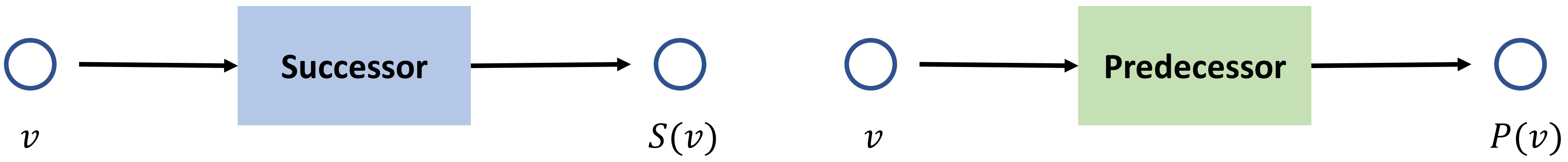
End of the Line (EOL)



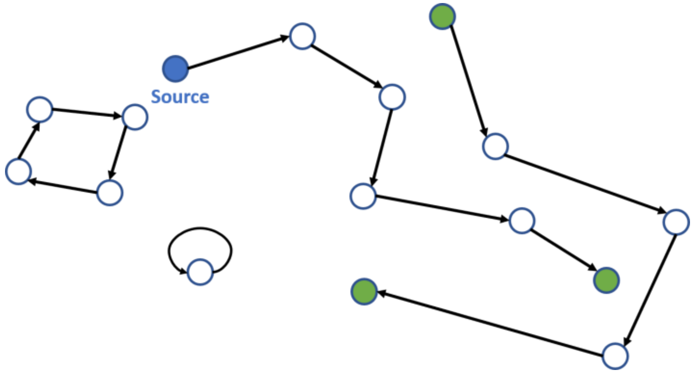


End of the Line (EOL)





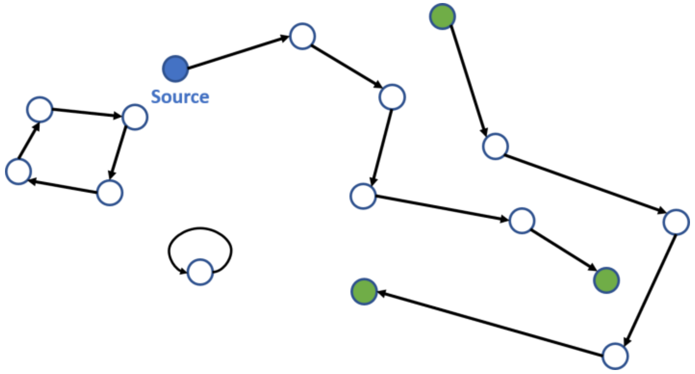
PPAD and NASH [Papadimitriou’94]



Reduction

	Left	Right
Left	1 \ -1	-1\ 1
Right	-1 \ 1	1 \ -1

PPAD and NASH [Papadimitriou’94]

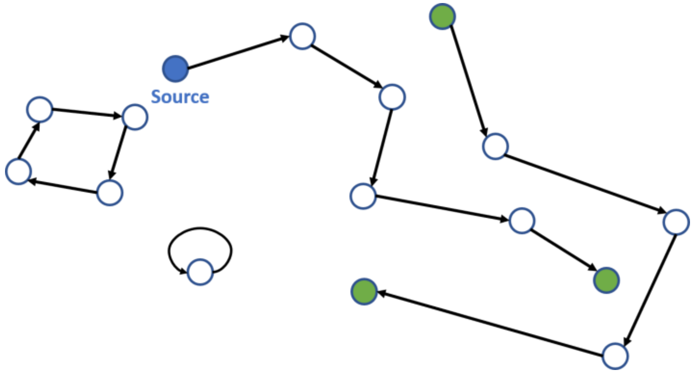


Reduction

Reduction

	Left	Right
Left	1 \ -1	-1\ 1
Right	-1 \ 1	1 \ -1

PPAD and NASH [Papadimitriou’94]

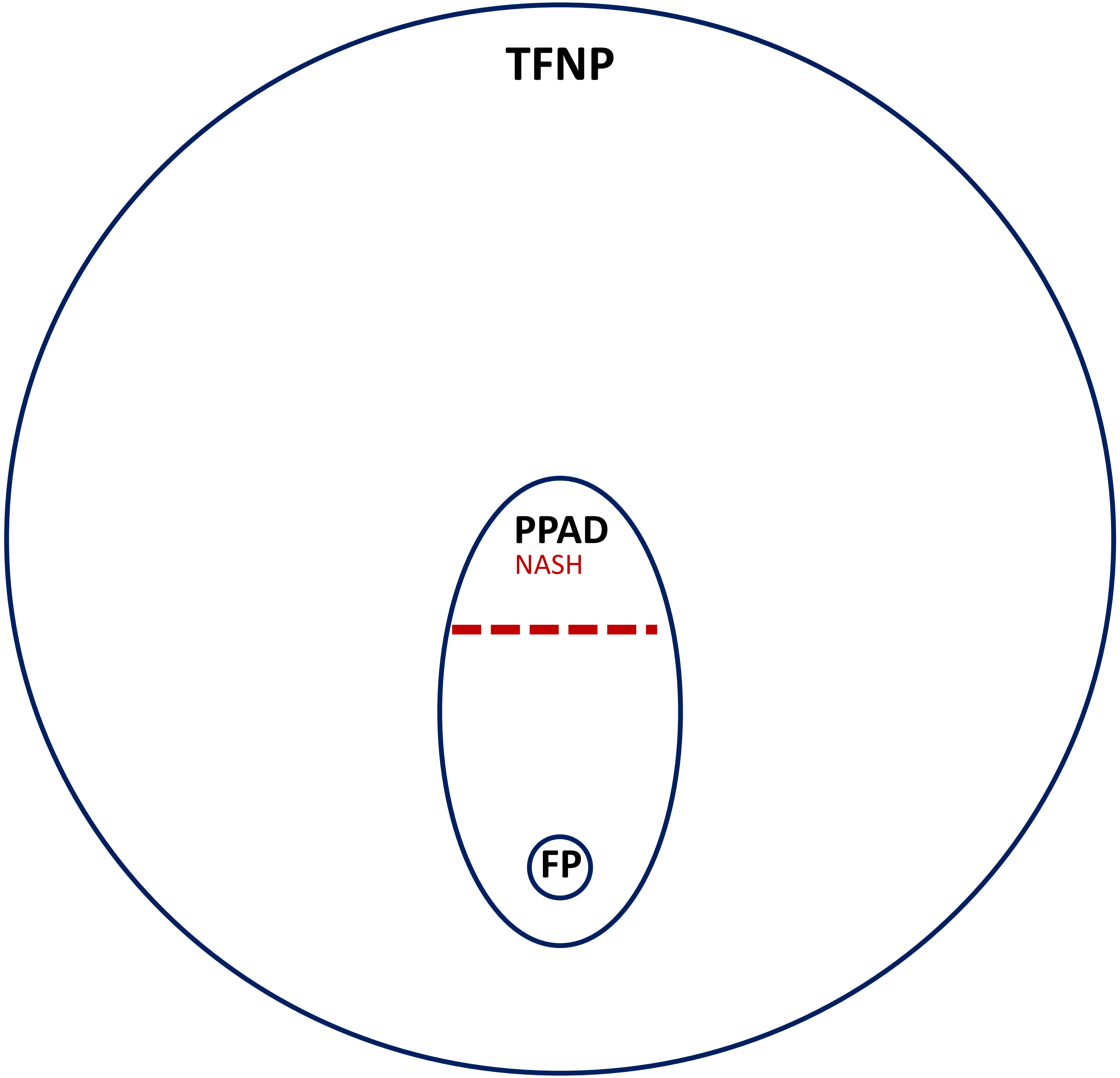


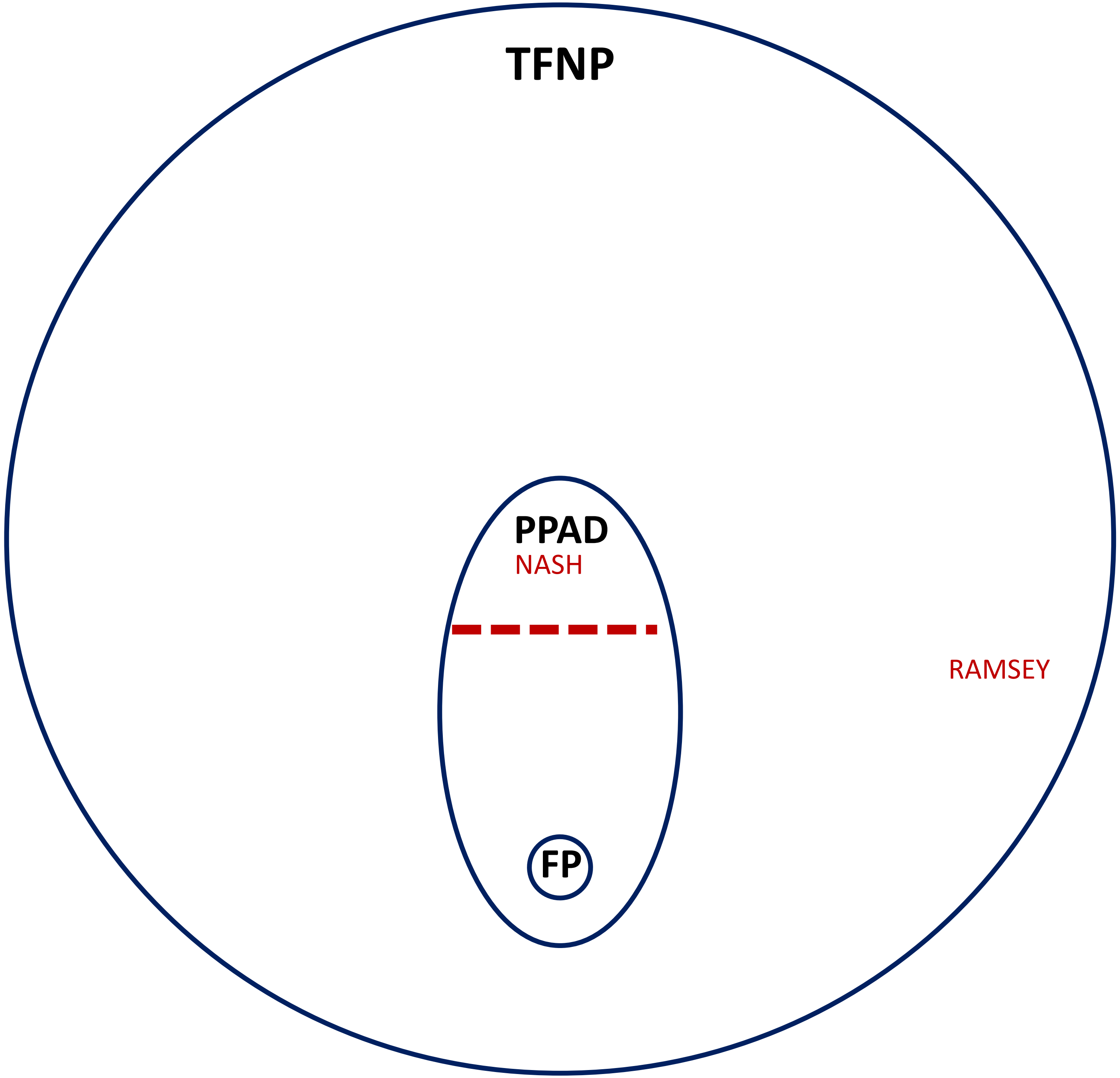
Reduction

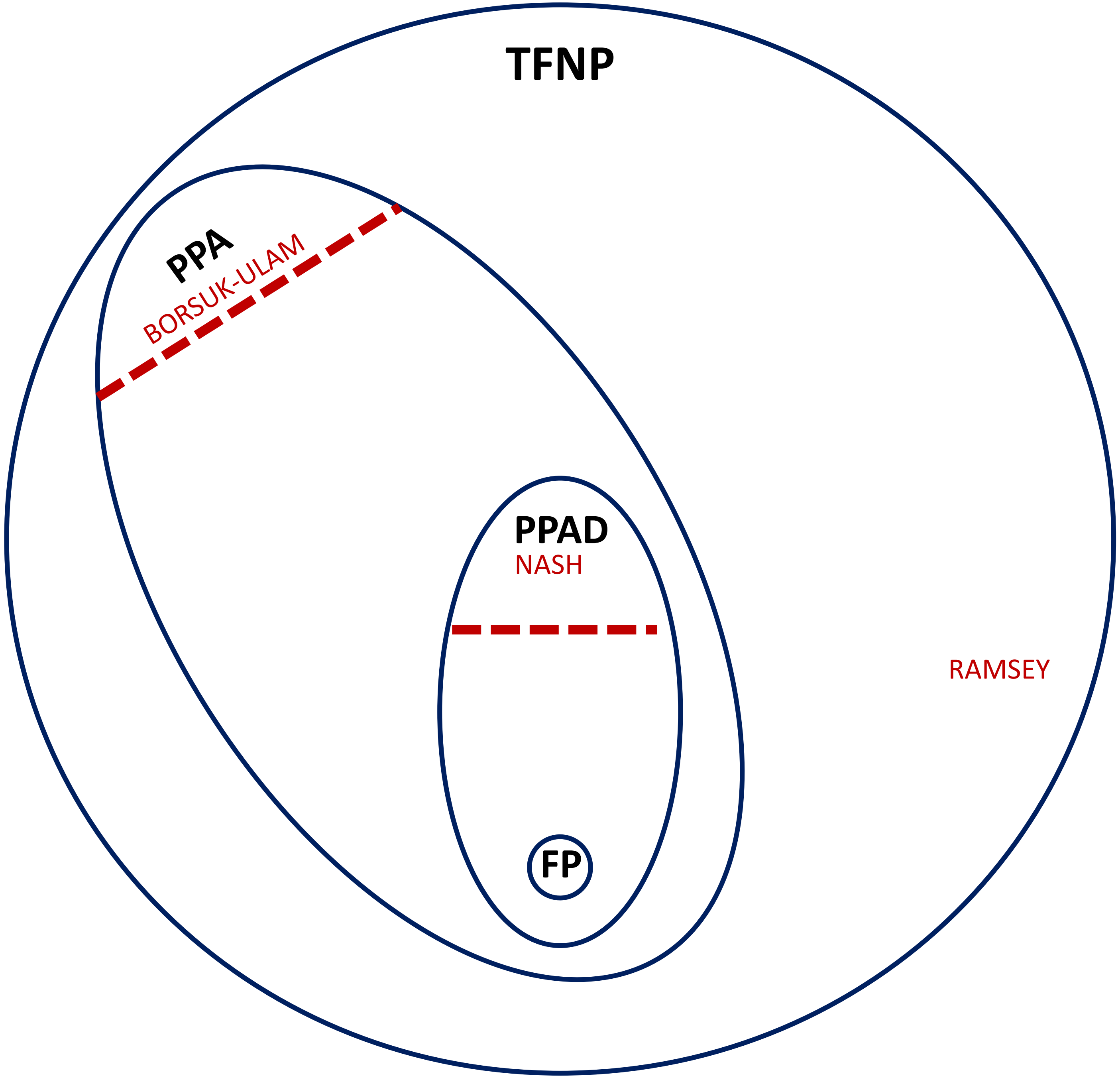
Reduction

	Left	Right
Left	1 \ -1	-1\ 1
Right	-1 \ 1	1 \ -1

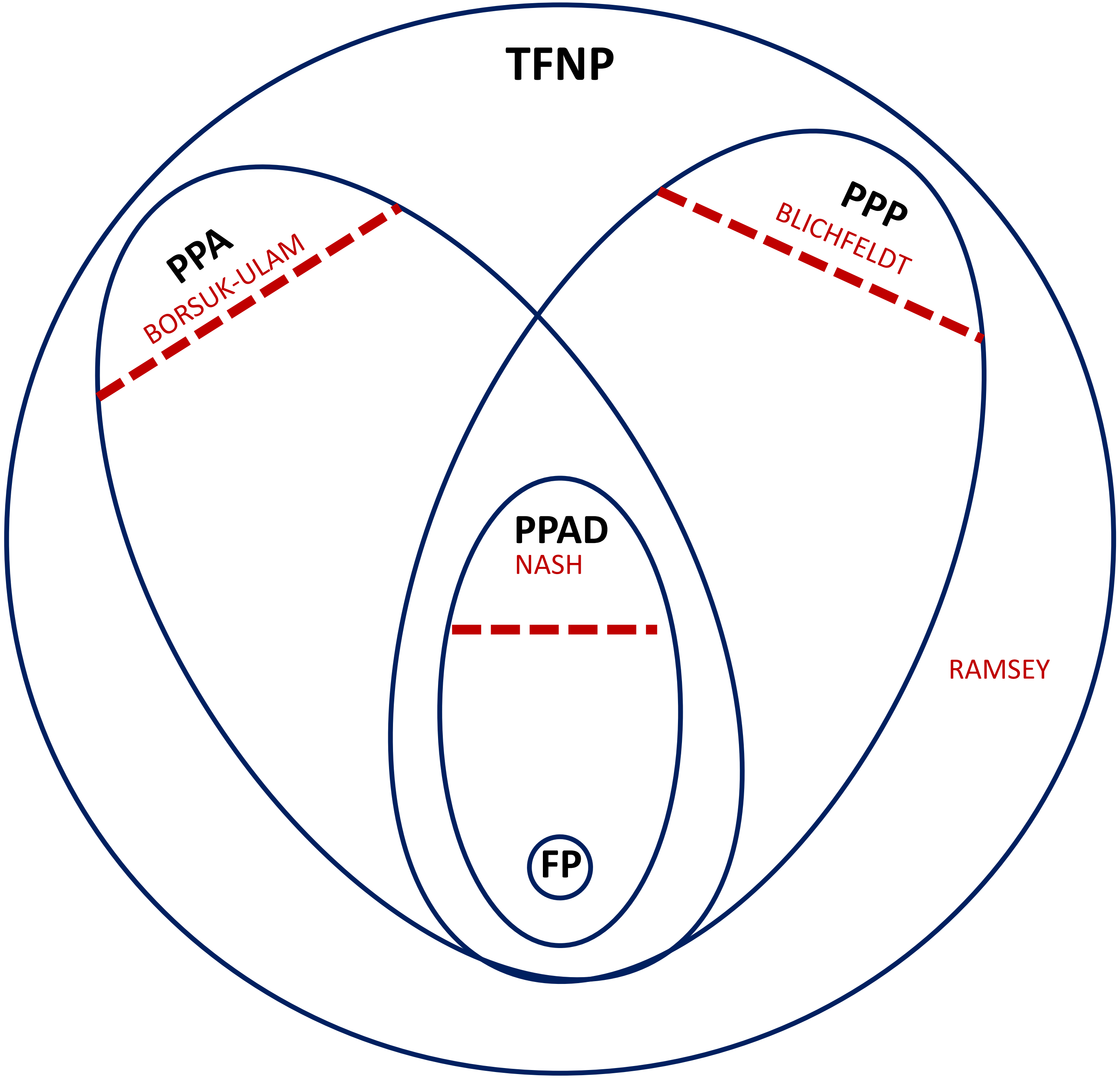
[Daskalakis-Goldberg-Papadimitriou 05],
[Chen-Deng 05]

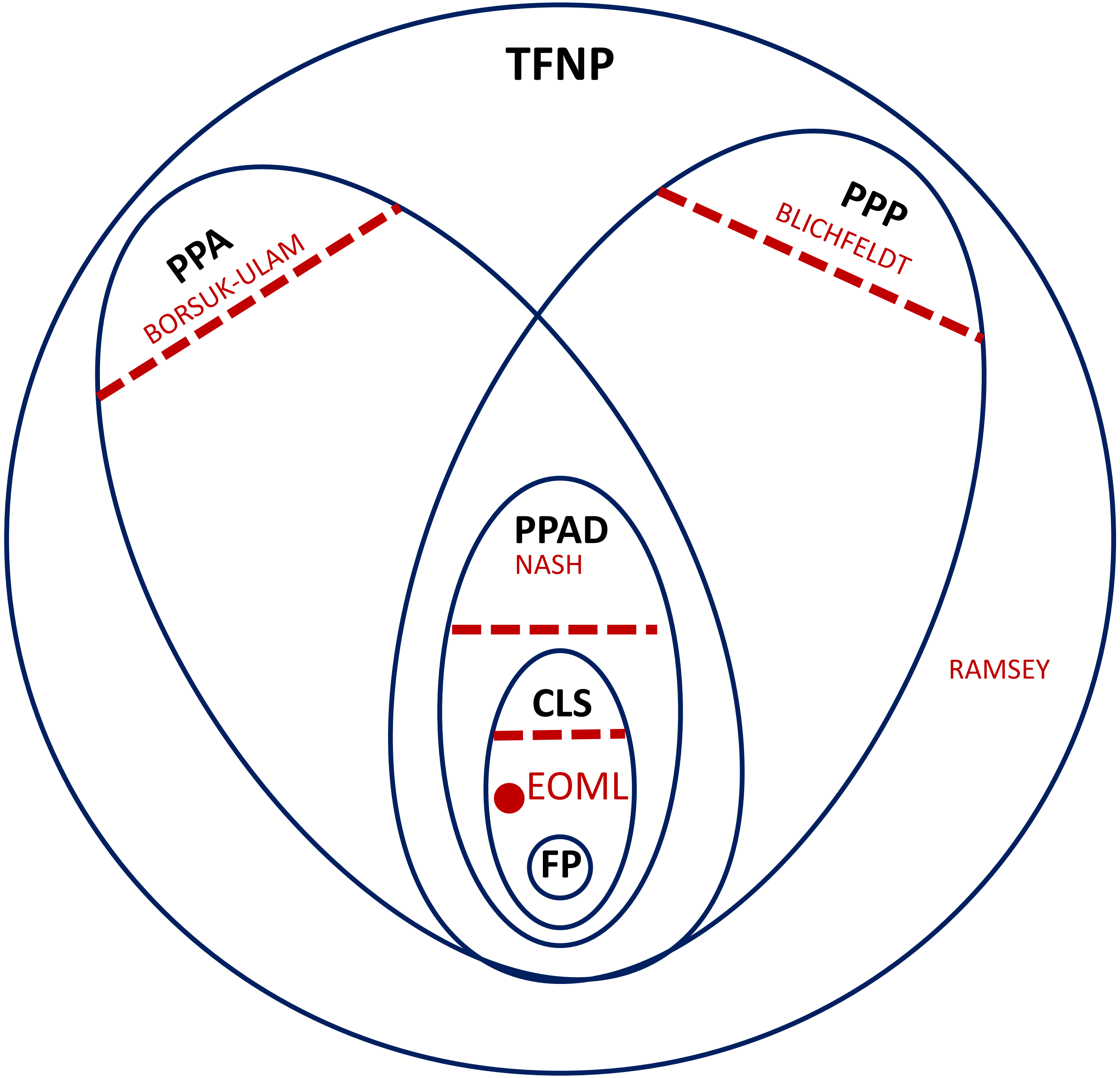

TFNP
PPAD
NASH
FP


TFNP
PPAD
NASH
FP
RAMSEY

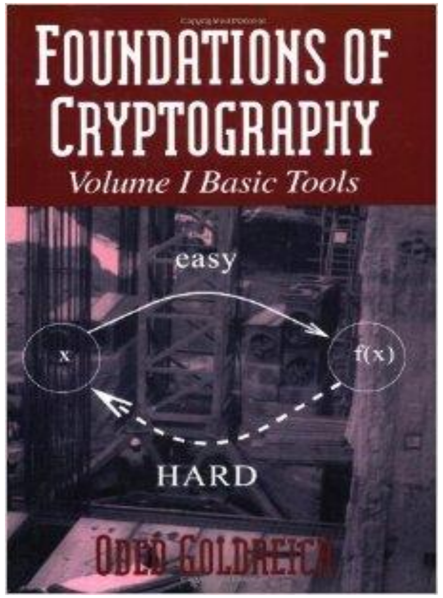


TFNP
PPA
BORSUK-ULAM
PPAD
NASH
FP
RAMSEY

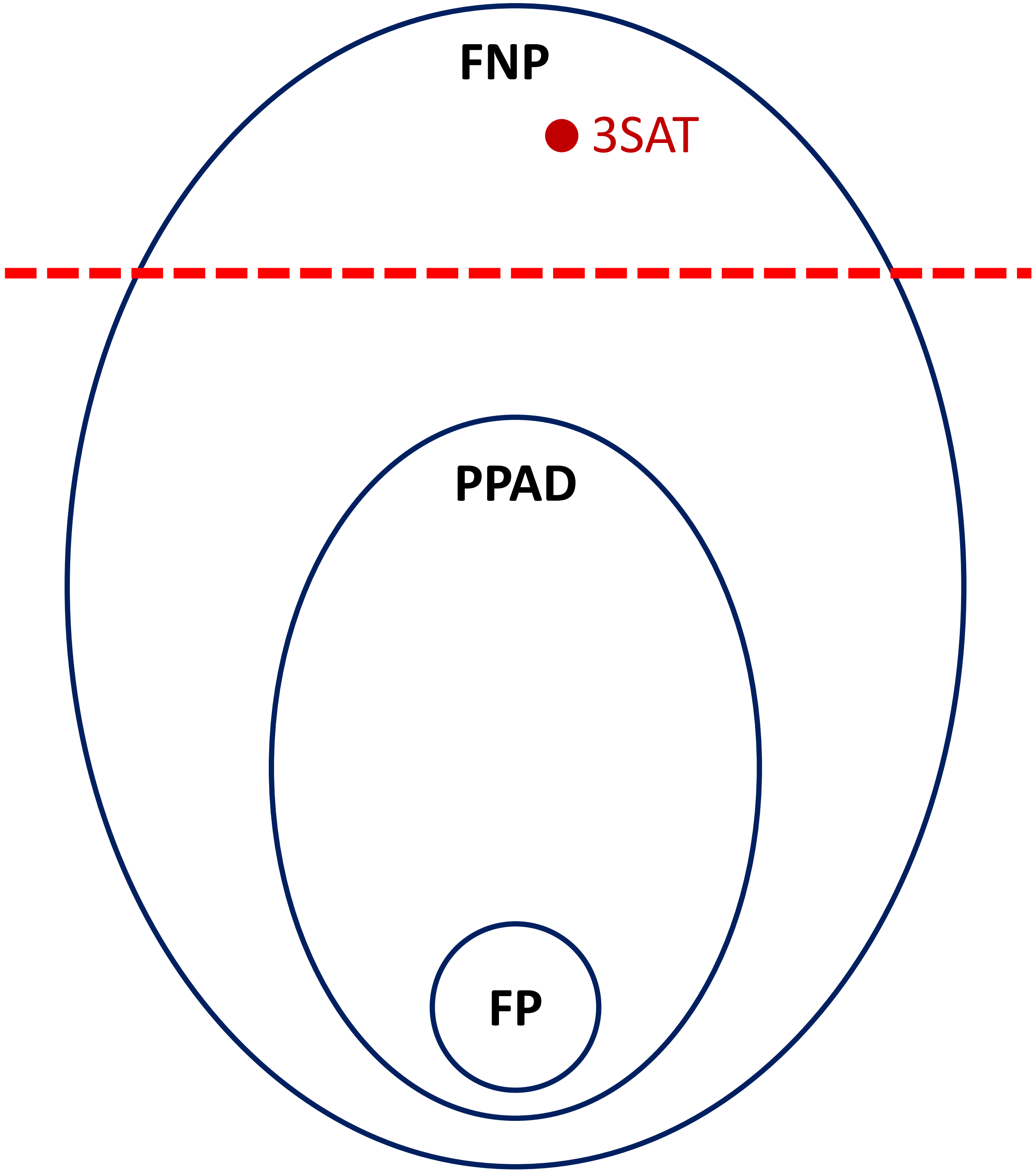

TFNP
PPA
BORSUK-ULAM
PPP
BLICHFELDT
PPAD
NASH
FP
RAMSEY

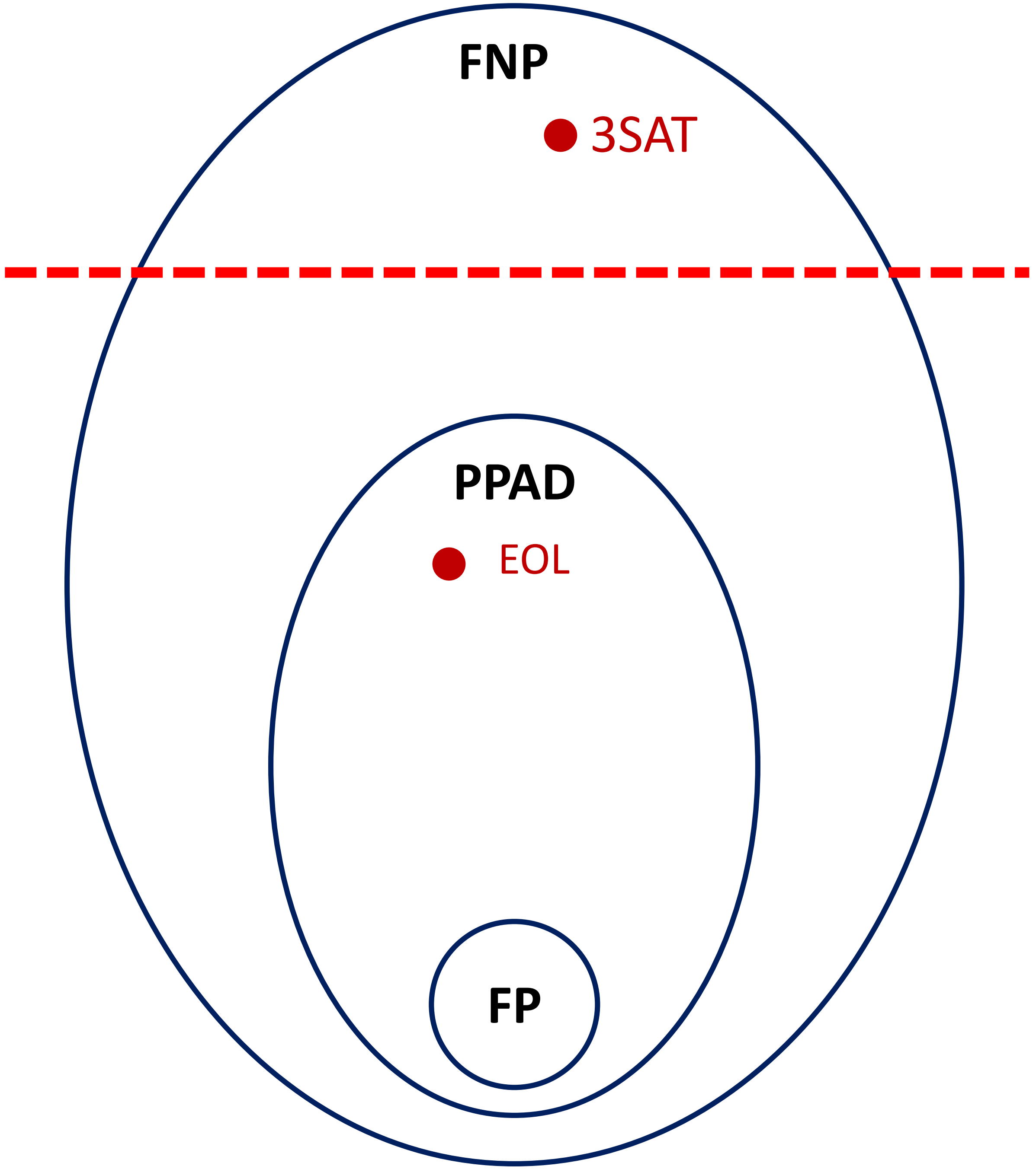

TFNP
PPA
BORSUK-ULAM
PPP
BLICHFELDT
PPAD
NASH
CLS
EOML
FP
RAMSEY

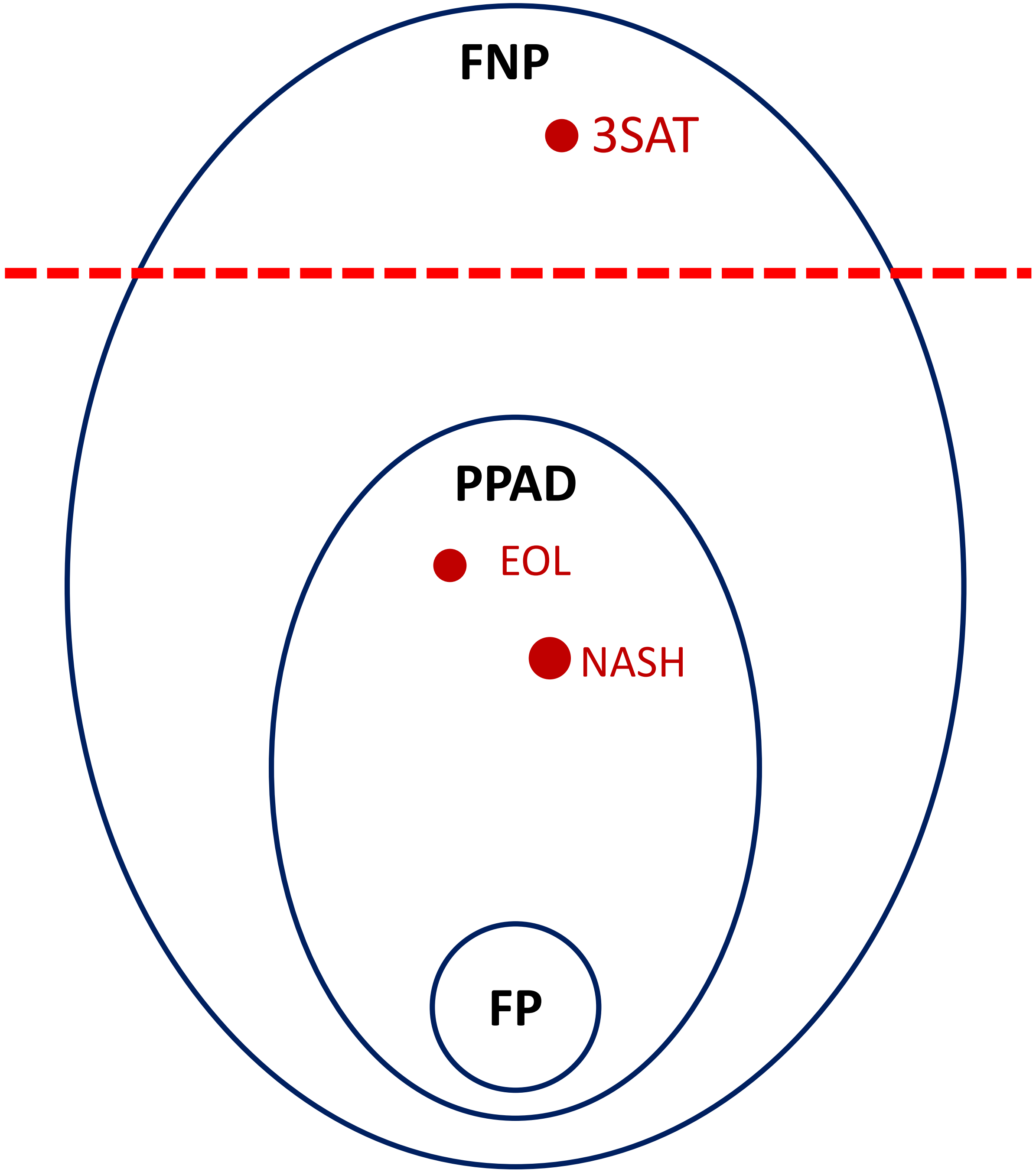
The Story Line

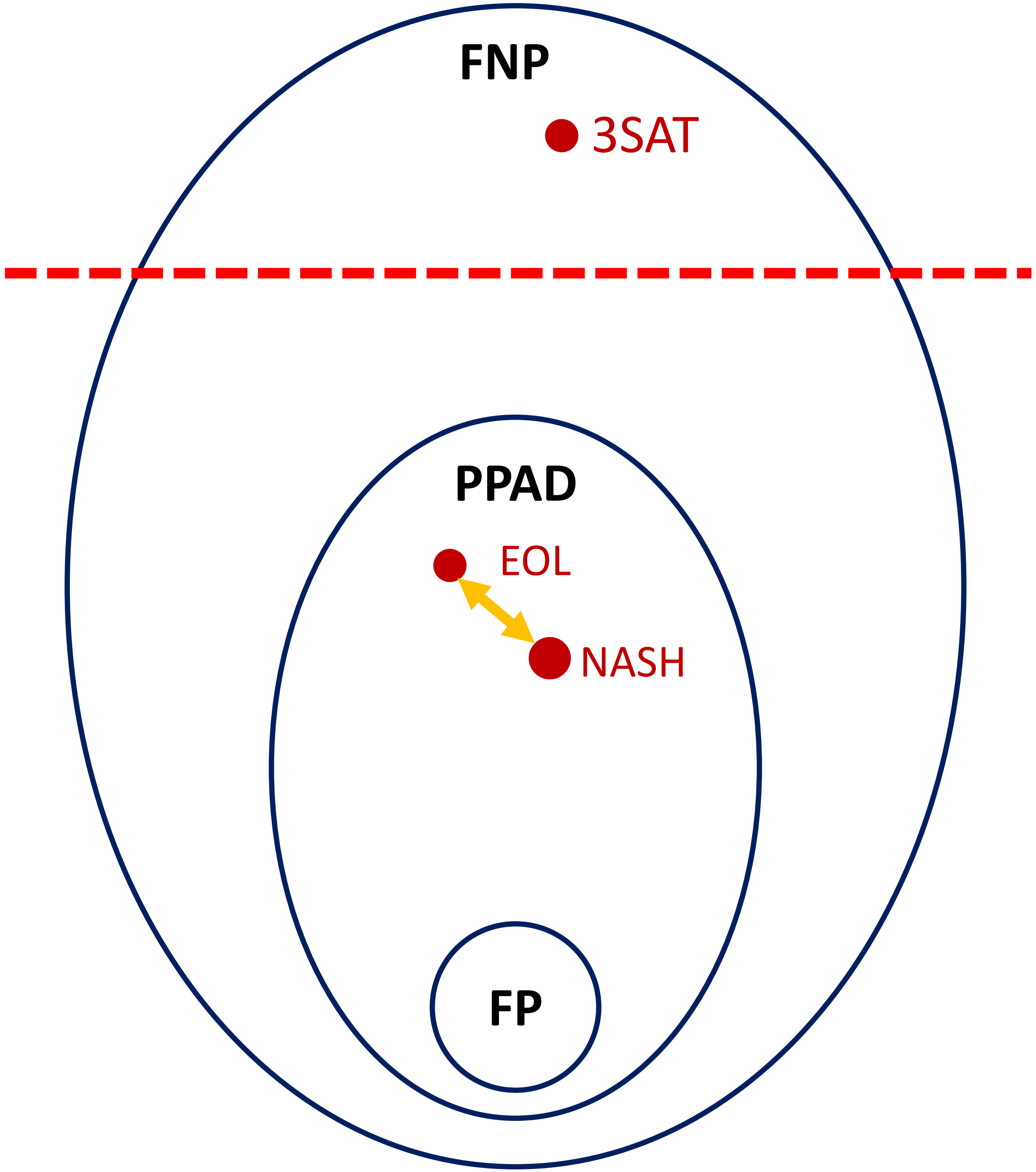


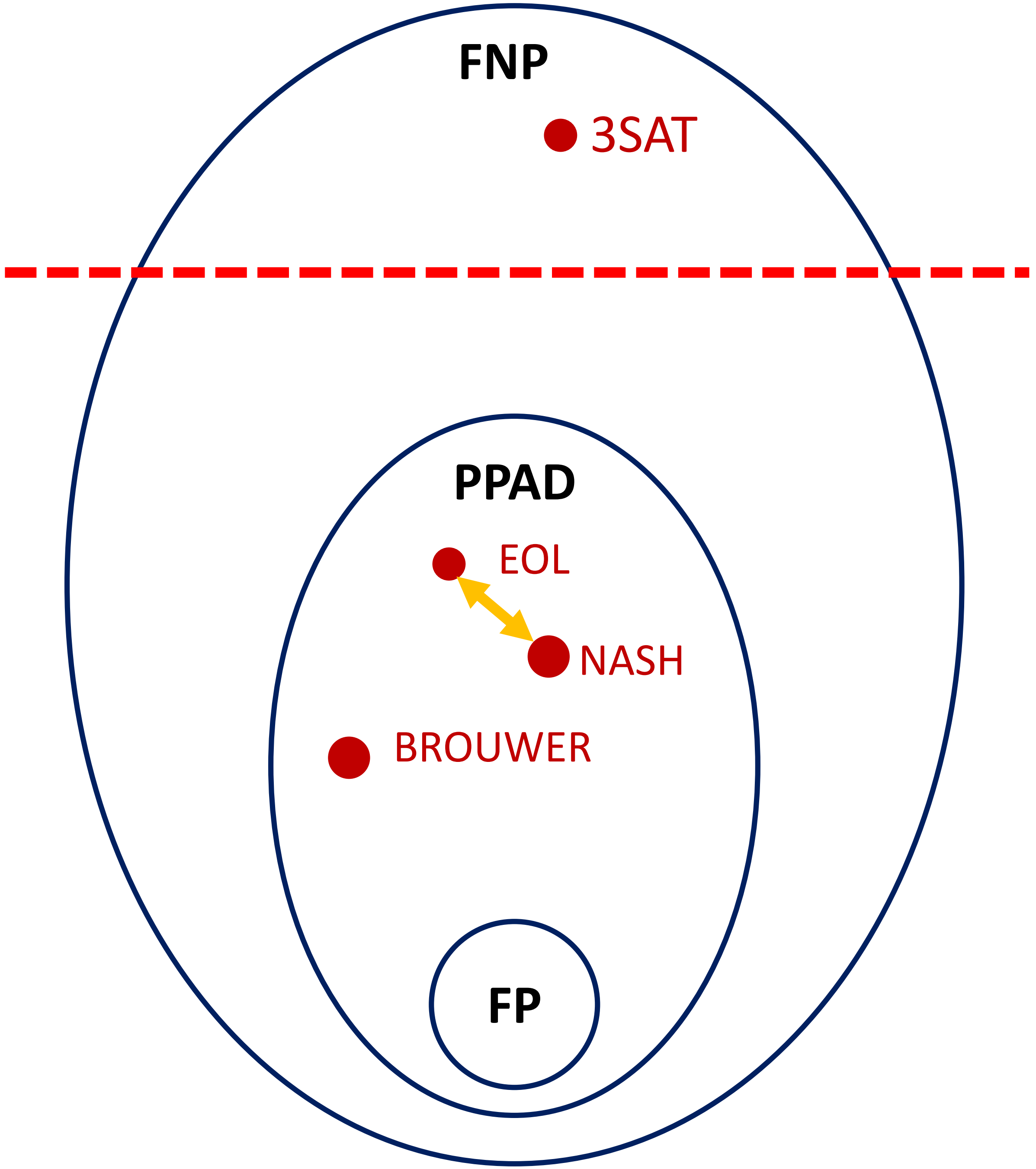
Games → Complexity → **Crypto**

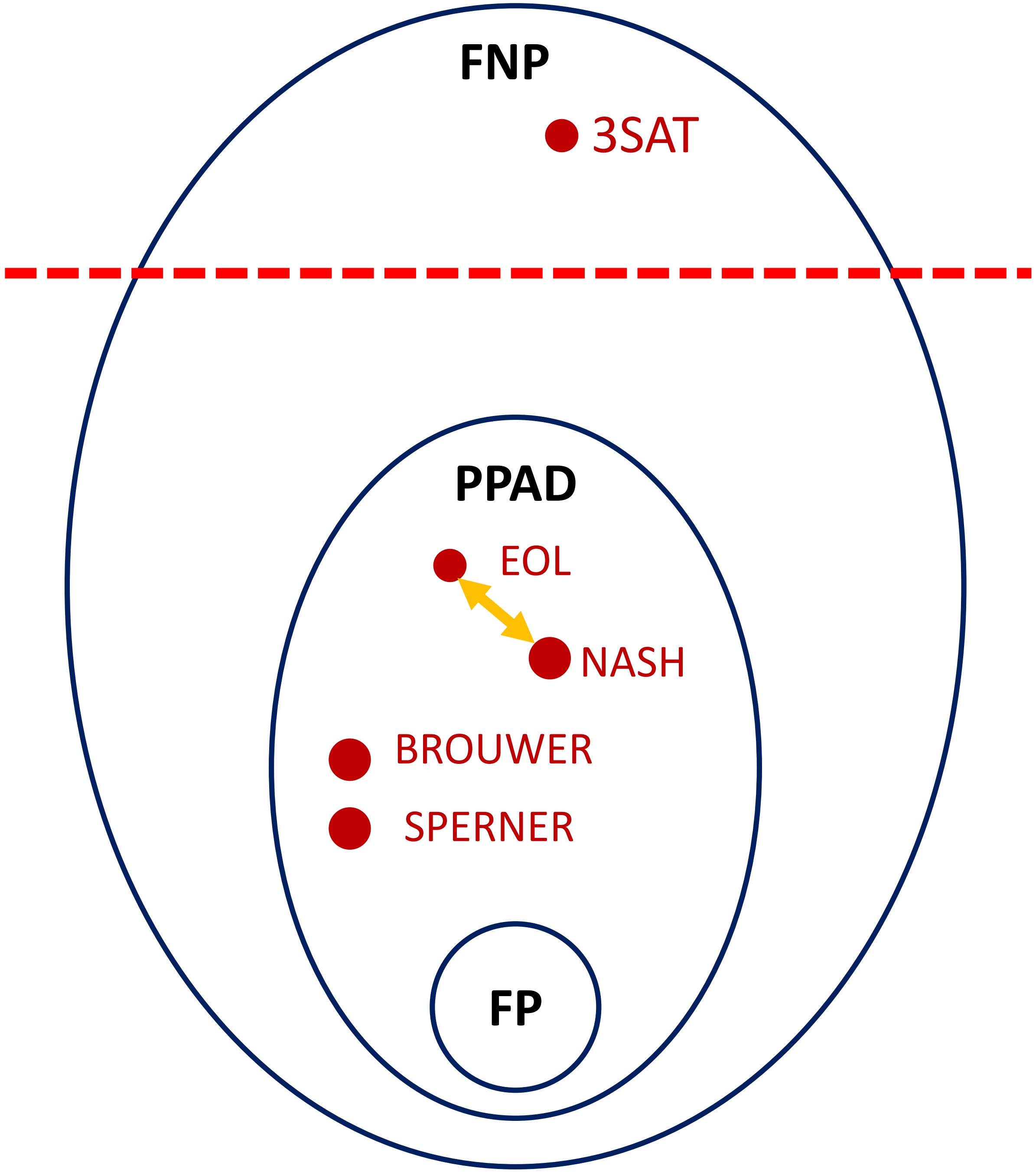

FNP
3SAT
PPAD
FP

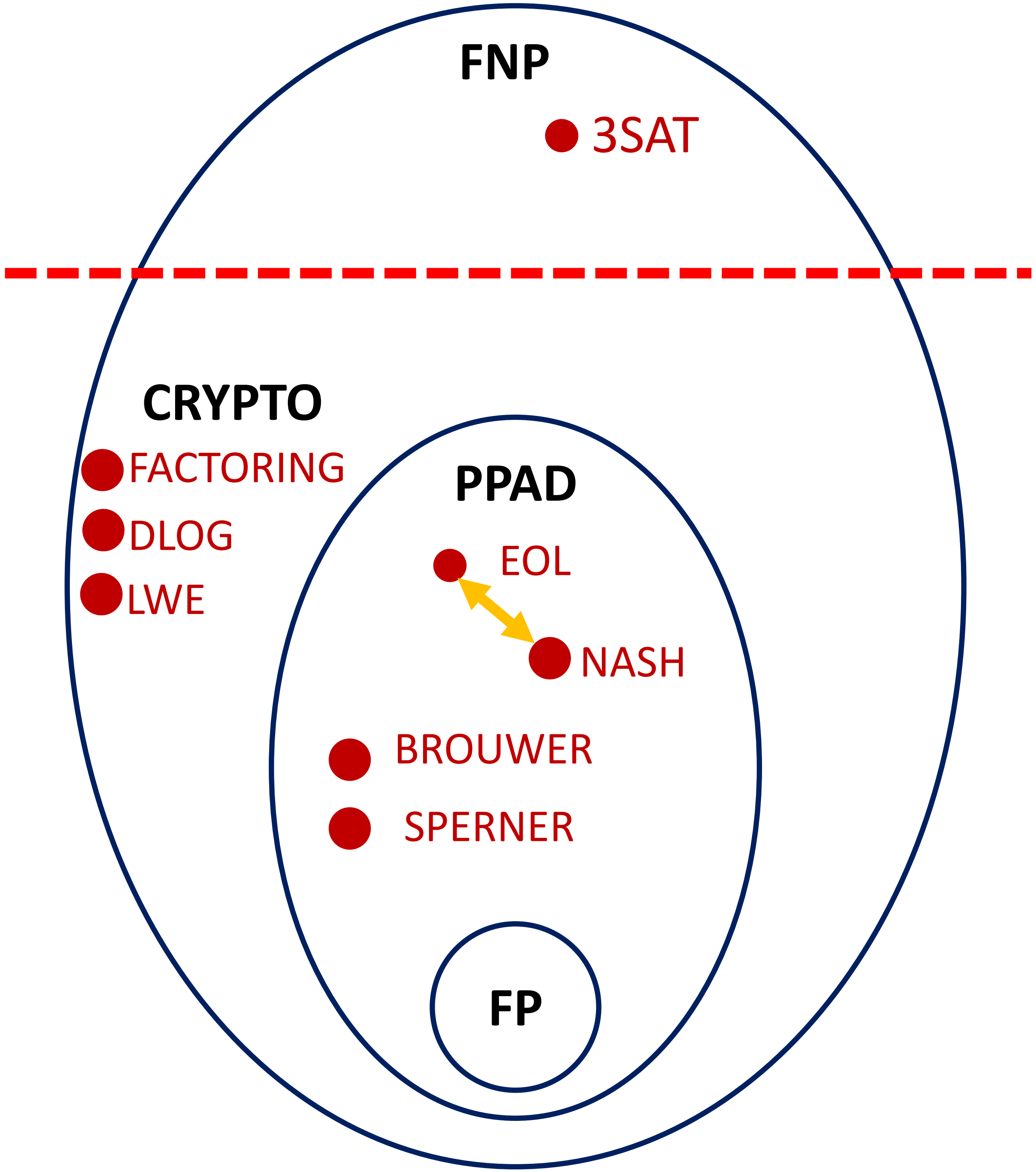

FNP
3SAT
PPAD
EOL
FP

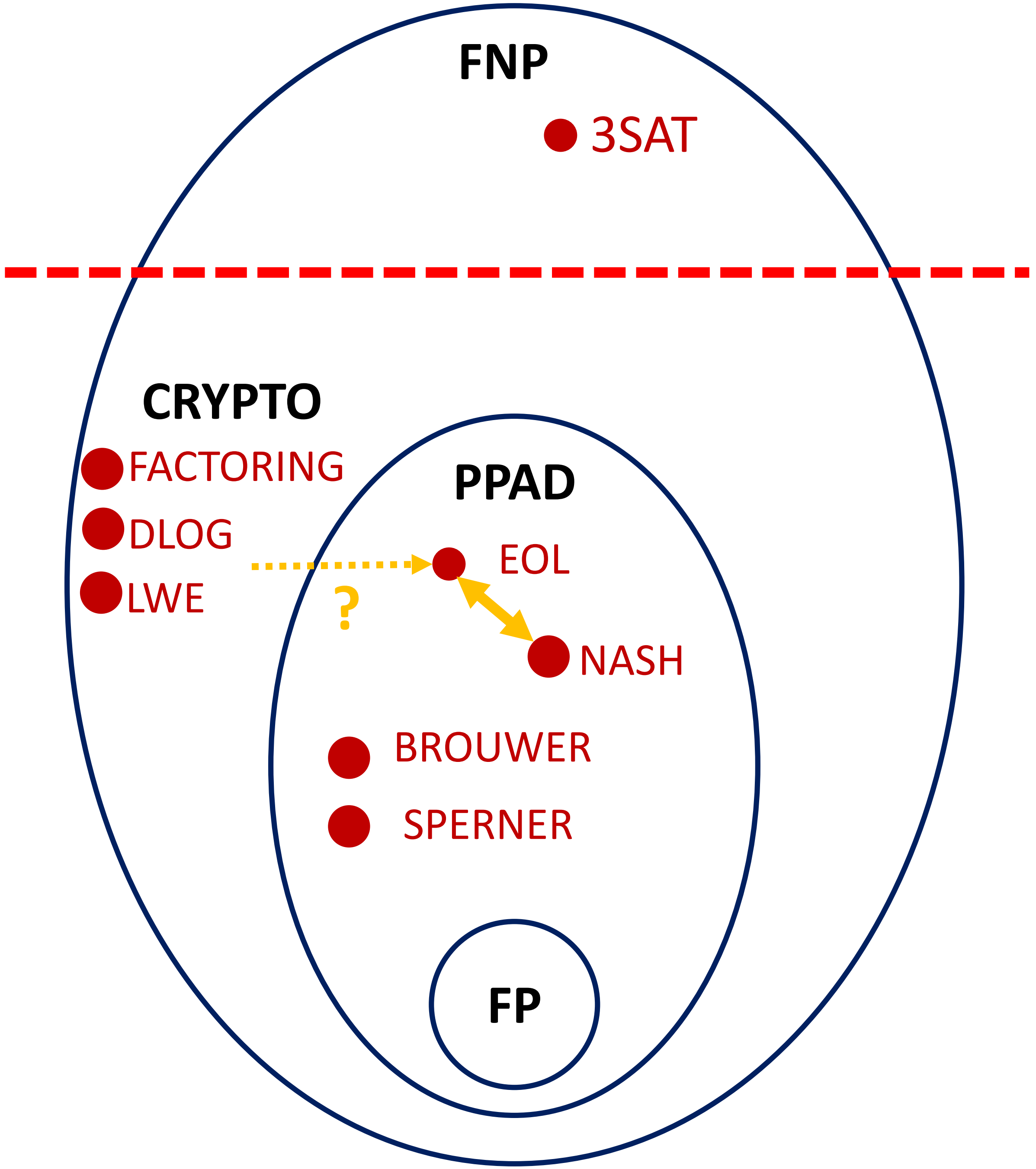

FNP
3SAT
PPAD
EOL
NASH
FP

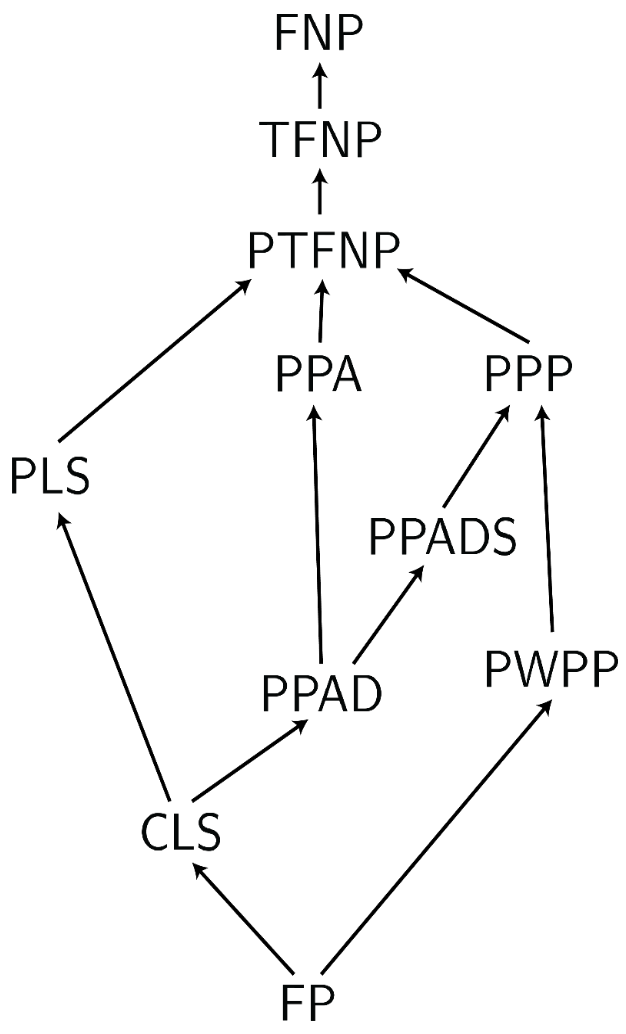

FNP
3SAT
PPAD
EOL
NASH
FP


FNP
3SAT
PPAD
EOL
NASH
BROUWER
FP


FNP
3SAT
PPAD
EOL
NASH
BROUWER
SPERNER
FP

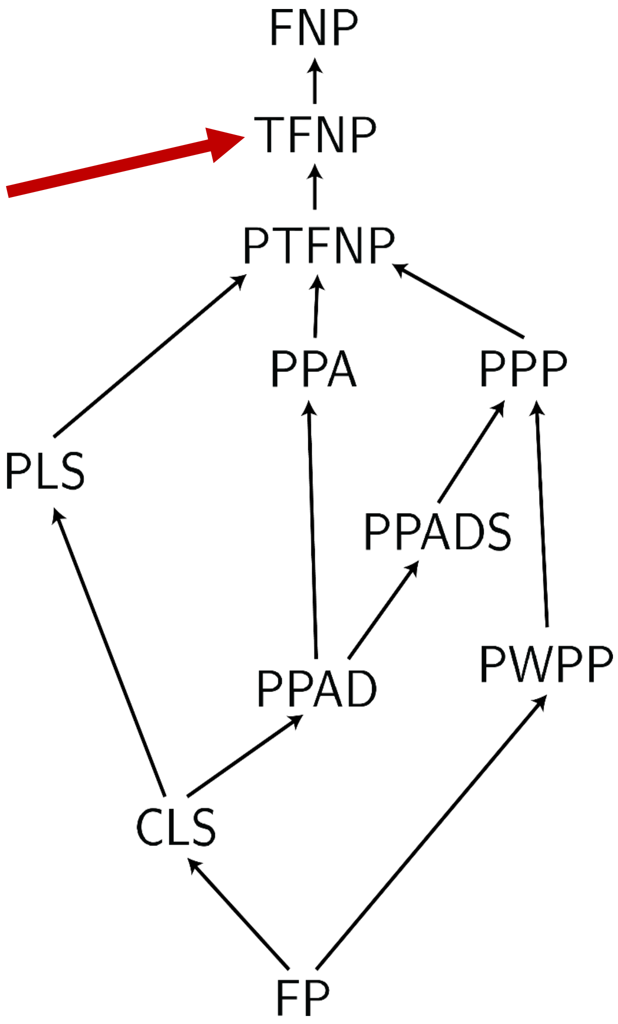

FNP
3SAT
CRYPTO
FACTORING
DLOG
LWE
PPAD
EOL
NASH
BROUWER
SPERNER
FP

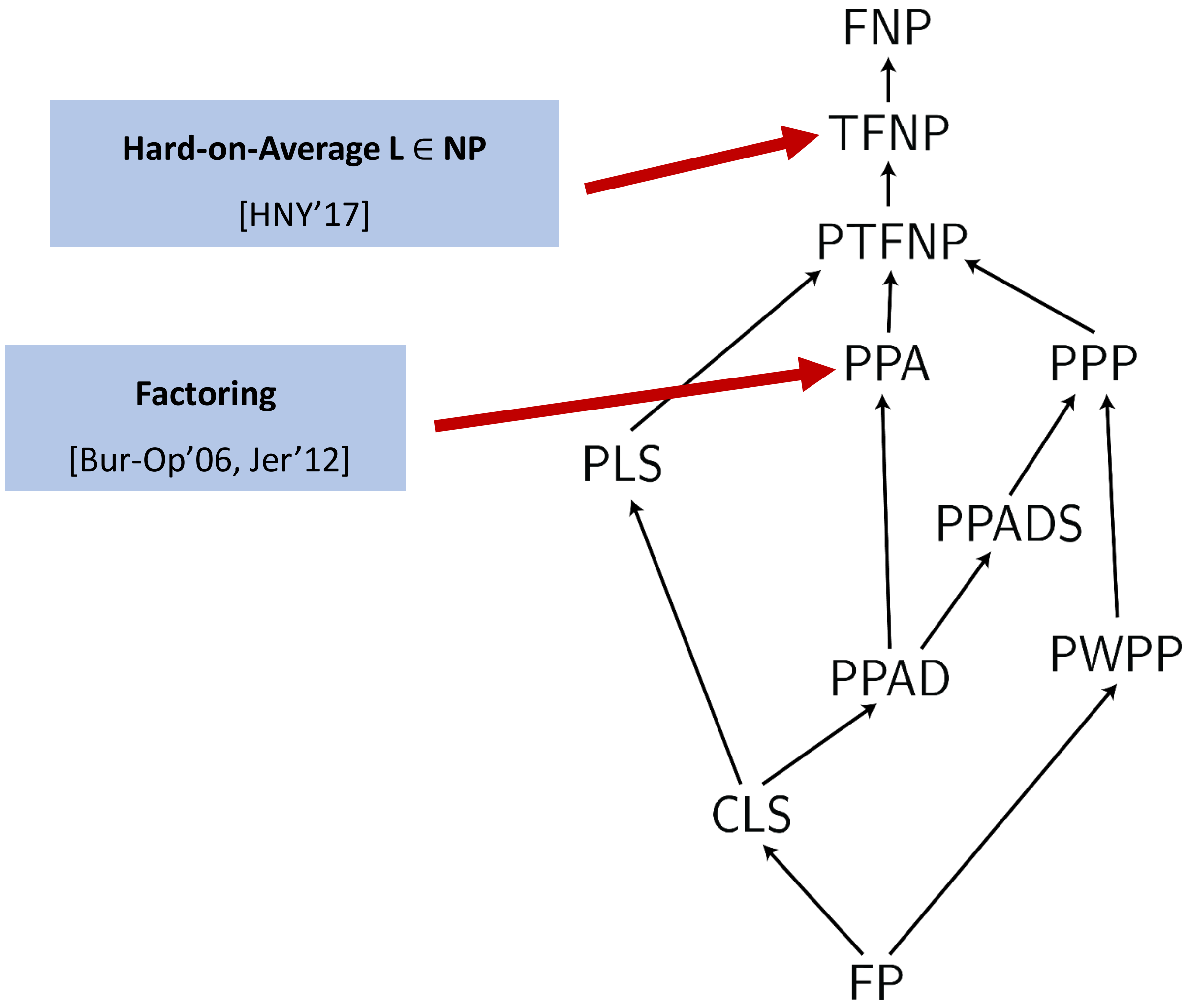

FNP
3SAT
CRYPTO
FACTORING
DLOG
LWE
PPAD
EOL
?
NASH
BROUWER
SPERNER
FP

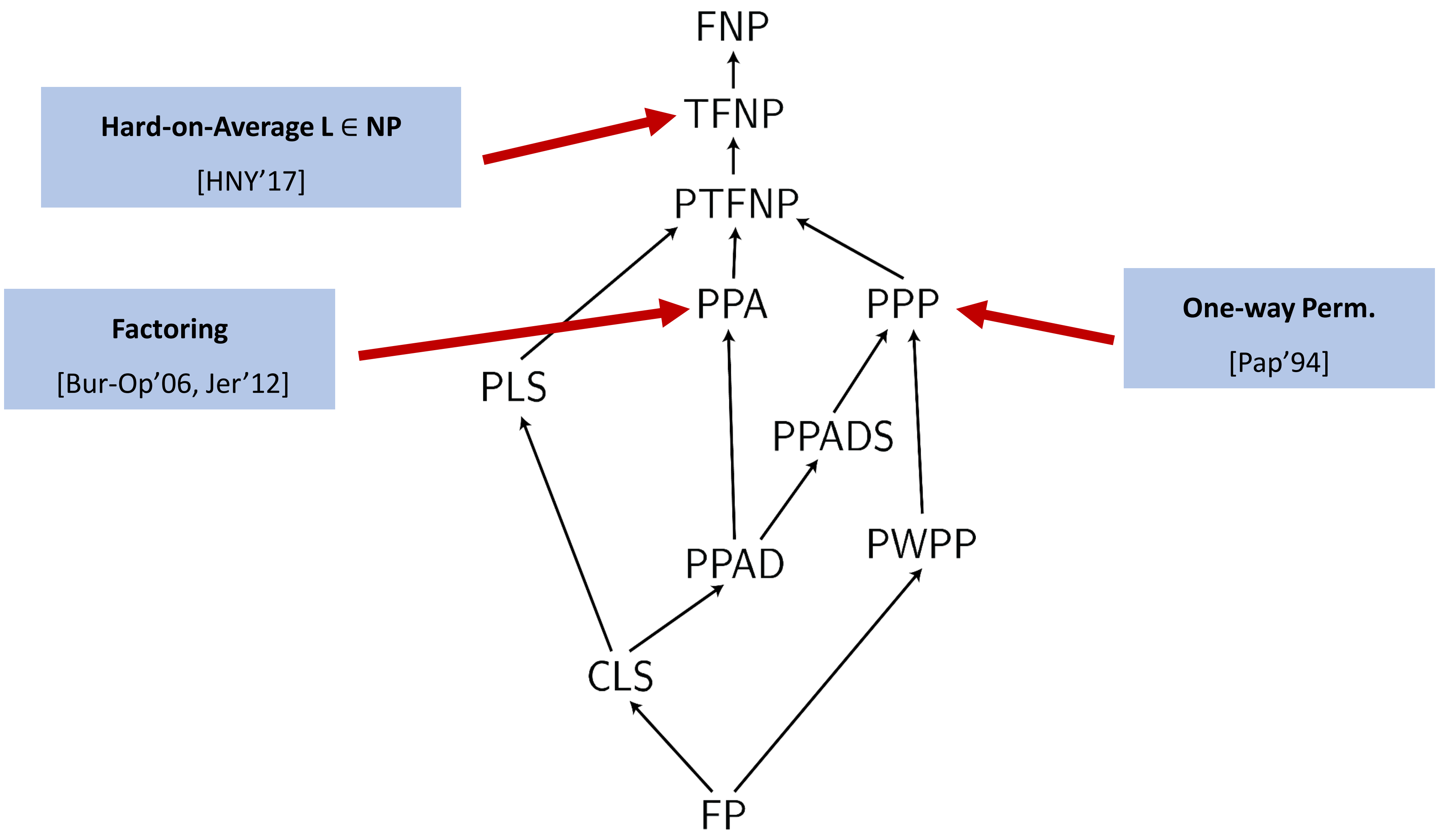

FNP
TFNP
PTFNP
PPA
PPP
PLS
PPADS
PPAD
PWPP
CLS
FP

Hard-on-Average L ∈ NP

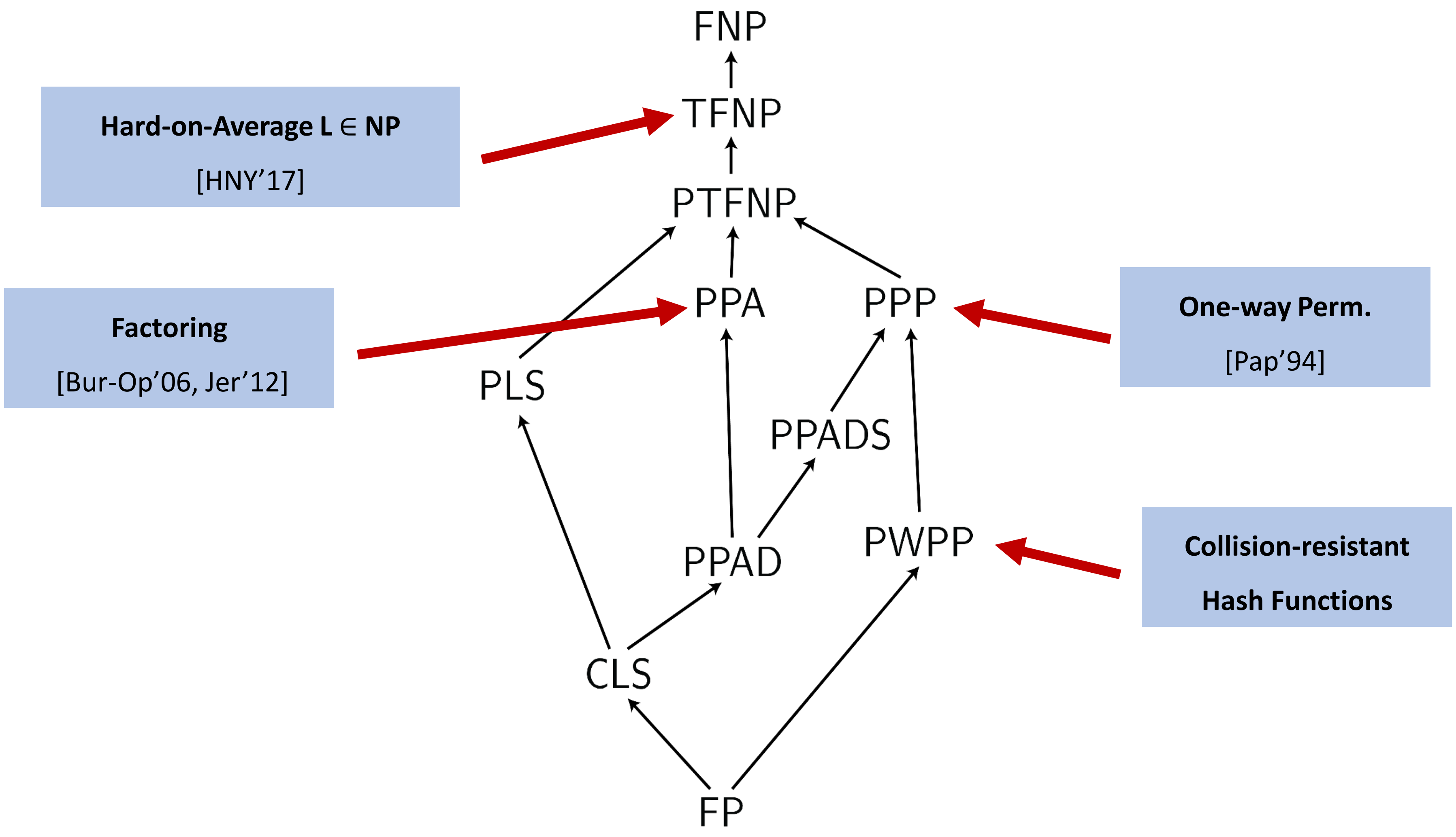
[HNY'17]



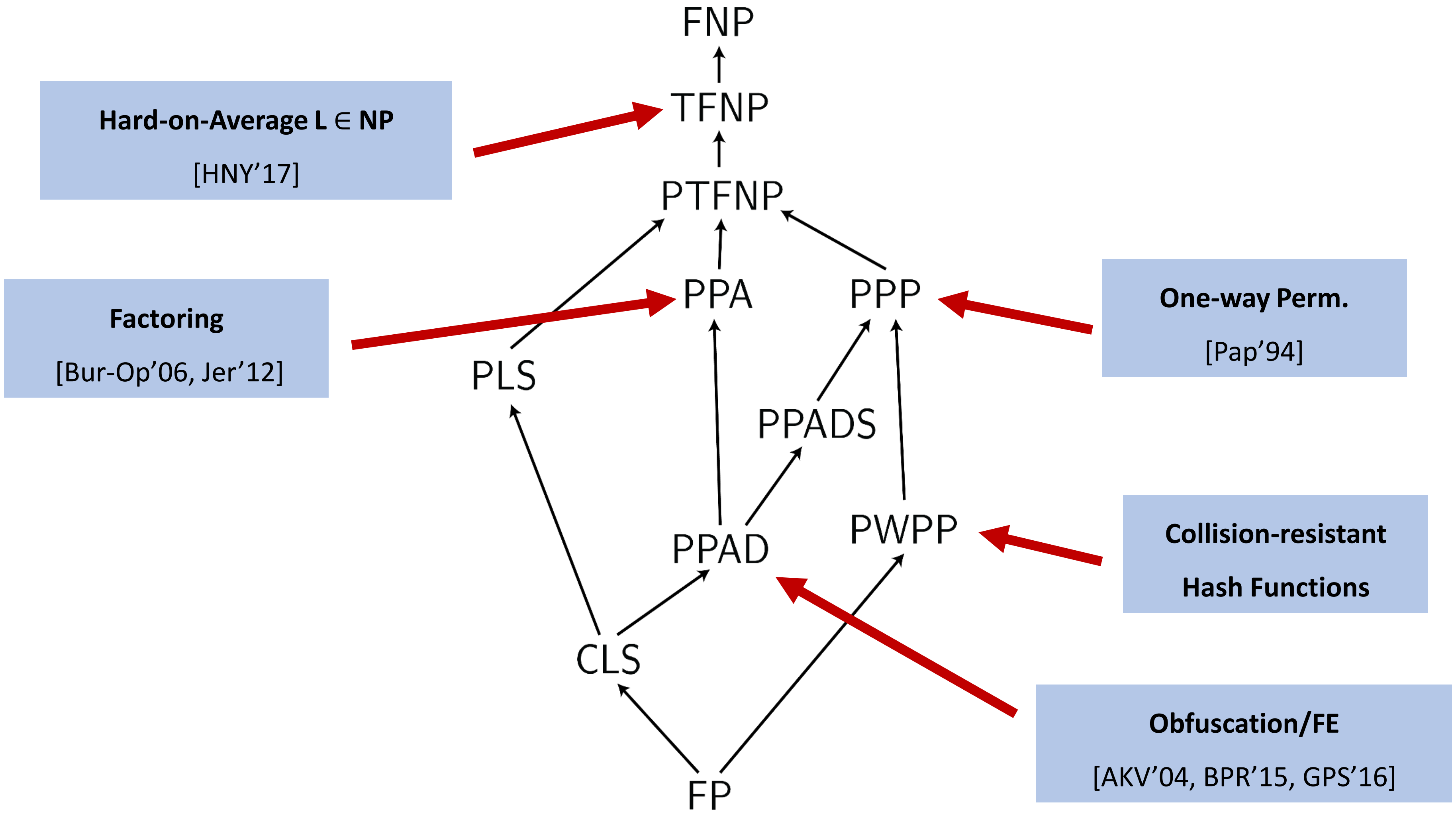

Hard-on-Average L ∈ NP
[HNY'17]
Factoring
[Bur-Op'06, Jer'12]
FNP
TFNP
PTFNP
PPA
PPP
PLS
PPADS
PPAD
PWPP
CLS
FP

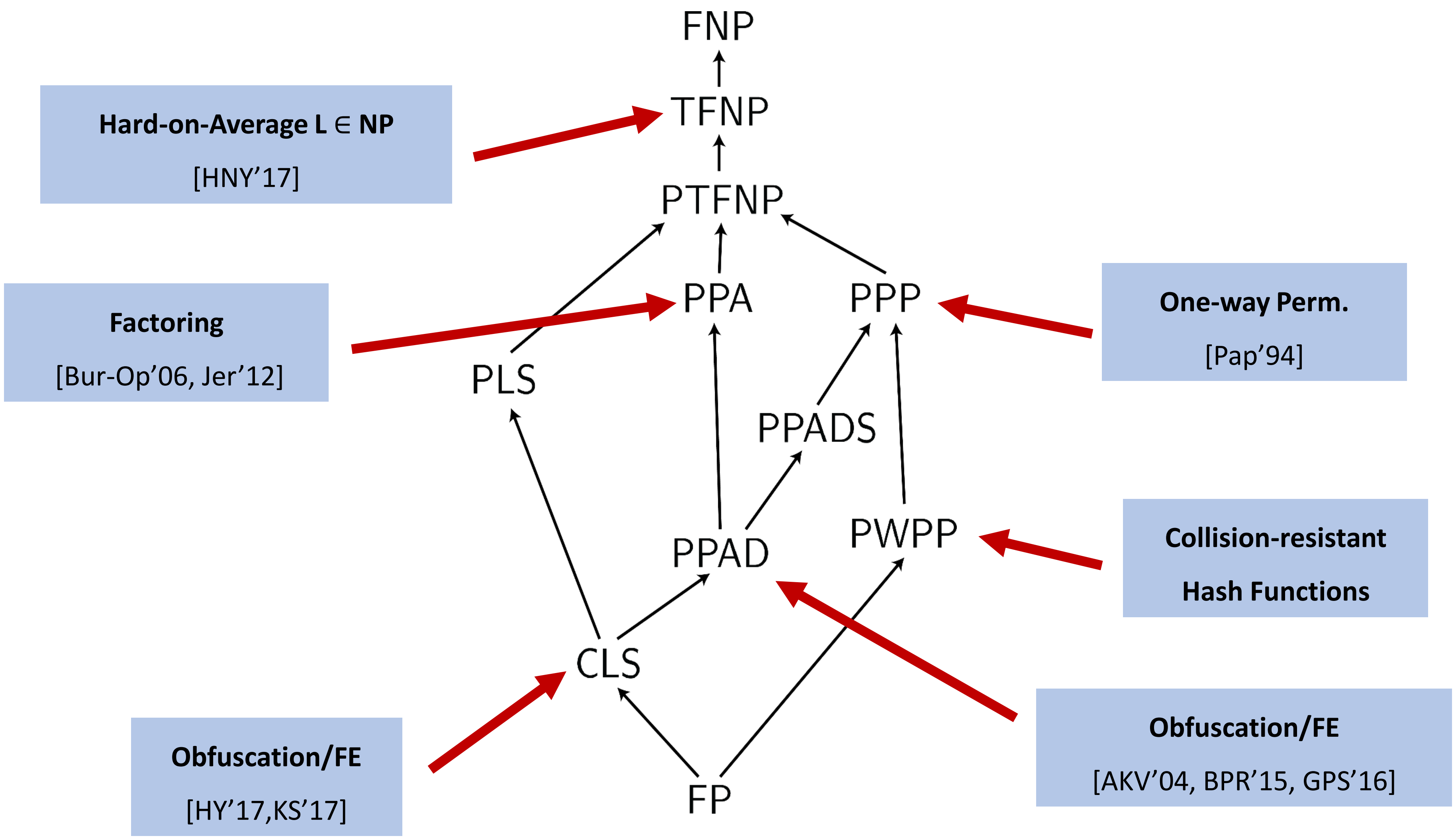


FNP
TFNP
PTFNP
PPA
PPP
PLS
PPADS
PPAD
PWPP
CLS
FP
Hard-on-Average L ∈ NP
[HNY'17]
Factoring
[Bur-Op'06, Jer'12]
One-way Perm.
[Pap'94]

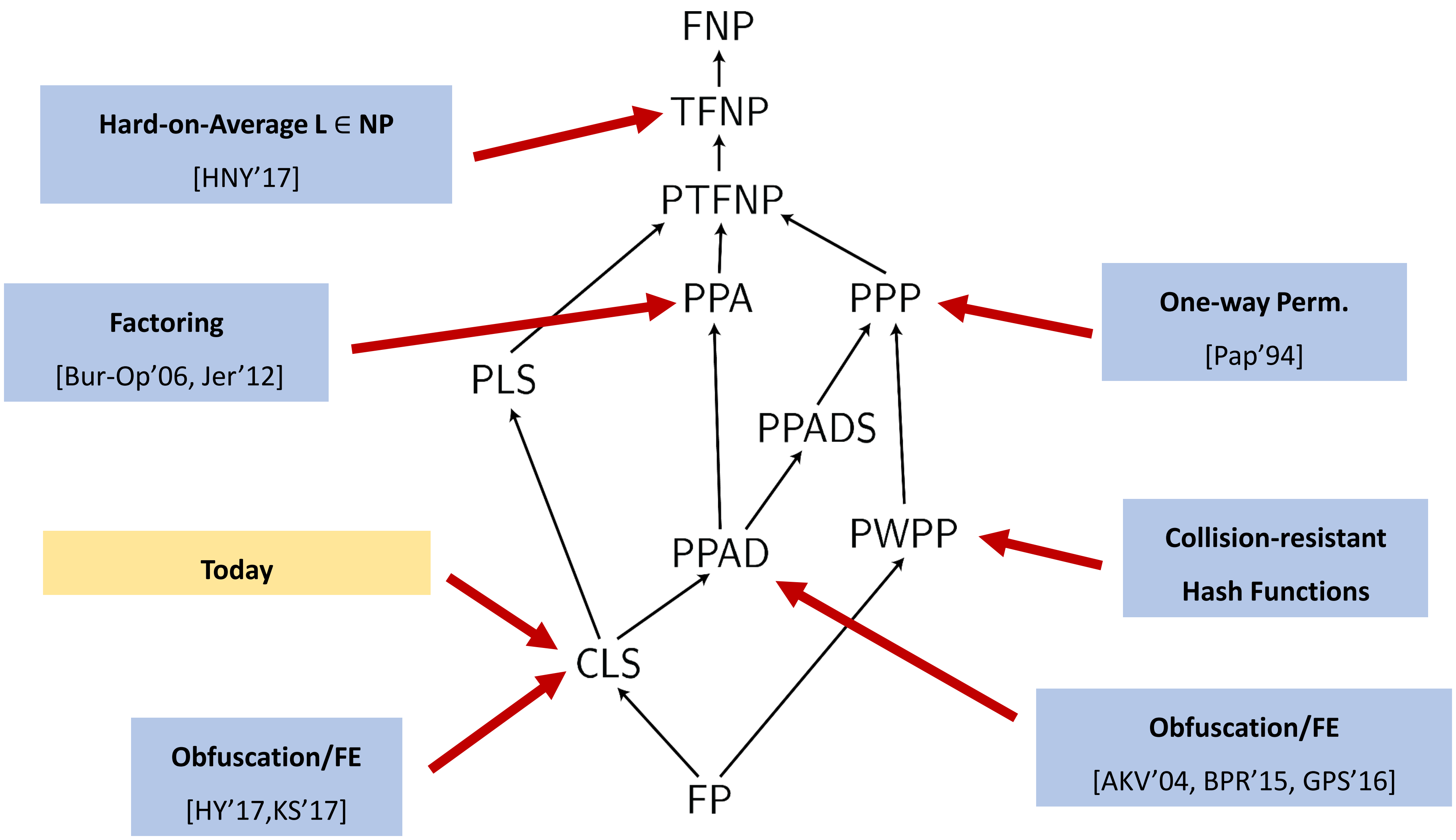


FNP
TFNP
PTFNP
PPA
PPP
PLS
PPADS
PPAD
PWPP
CLS
FP
Hard-on-Average L ∈ NP
[HNY'17]
Factoring
[Bur-Op'06, Jer'12]
One-way Perm.
[Pap'94]
Collision-resistant
Hash Functions


FNP
TFNP
PTFNP
PLS
PPA
PPP
PPADS
PPAD
PWPP
CLS
FP
Hard-on-Average L ∈ NP
[HNY'17]
Factoring
[Bur-Op'06, Jer'12]
One-way Perm.
[Pap'94]
Collision-resistant
Hash Functions
Obfuscation/FE
[AKV'04, BPR'15, GPS'16]

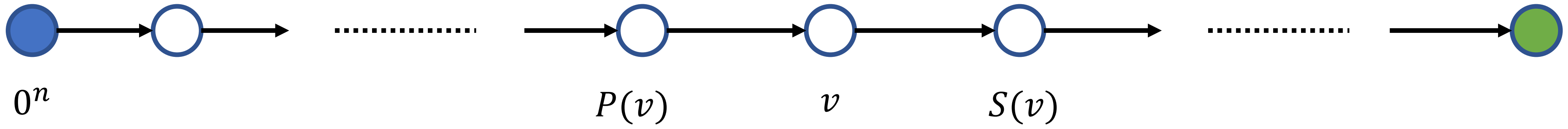


FNP
TFNP
PTFNP
PPA
PPP
PLS
PPADS
PPAD
PWPP
CLS
FP
Hard-on-Average L ∈ NP
[HNY'17]
Factoring
[Bur-Op'06, Jer'12]
One-way Perm.
[Pap'94]
Collision-resistant
Hash Functions
Obfuscation/FE
[HY'17,KS'17]
Obfuscation/FE
[AKV'04, BPR'15, GPS'16]


FNP
TFNP
PTFNP
PPA
PPP
PLS
PPADS
PPAD
PWPP
CLS
FP
Hard-on-Average L ∈ NP
[HNY'17]
Factoring
[Bur-Op'06, Jer'12]
Today
Obfuscation/FE
[HY'17,KS'17]
One-way Perm.
[Pap'94]
Collision-resistant
Hash Functions
Obfuscation/FE
[AKV'04, BPR'15, GPS'16]

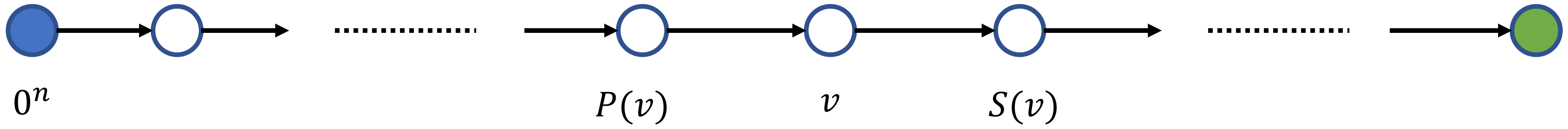
EOL Hardness from Obfuscation

[Bitansky-Paneth Rosen'15]



EOL Hardness from Obfuscation

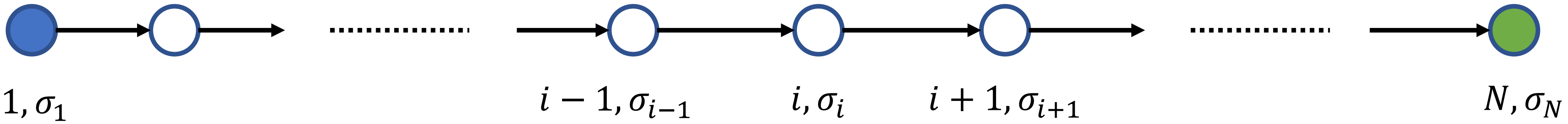
[Bitansky-Paneth Rosen'15]



PPAD/CLS hardness can be based on
indistinguishability obfuscation (iO)

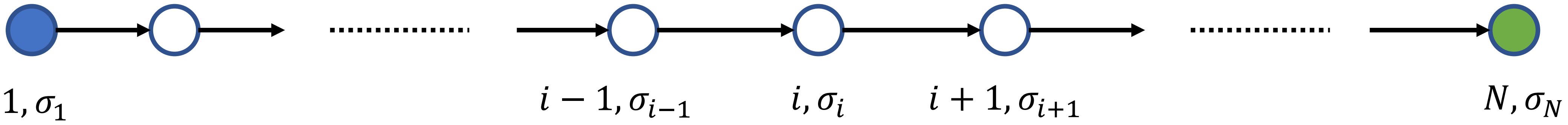
Sink of Verifiable Line (SVL)

[Abbott-Kane-Valiant’04,Bitansky-Paneth Rosen’15]



Sink of Verifiable Line (SVL)

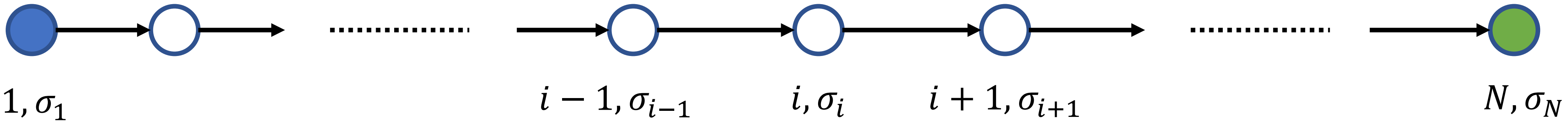
[Abbott-Kane-Valiant’04,Bitansky-Paneth Rosen’15]



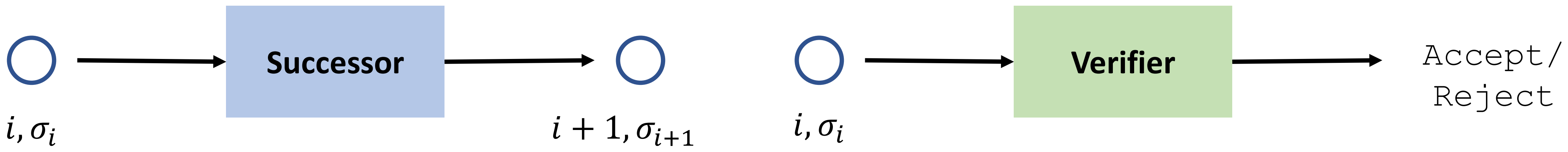
If the path is verifiable, then **Predecessor** is for free.

Sink of Verifiable Line (SVL)

[Abbott-Kane-Valiant'04,Bitansky-Paneth Rosen'15]

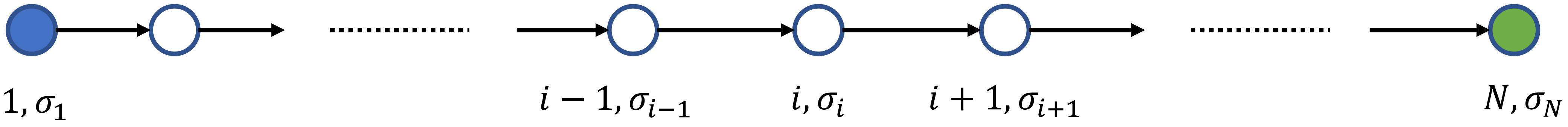


If the path is verifiable, then **Predecessor** is for free.



Sink of Verifiable Line (SVL)

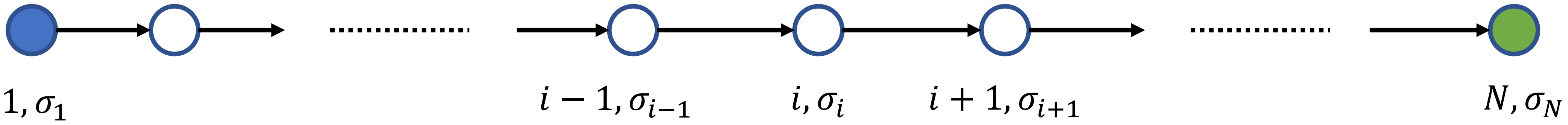
[Abbott-Kane-Valiant’04,Bitansky-Paneth Rosen’15]



If the path is verifiable, then **Predecessor** is for free.

Sink of Verifiable Line (SVL)

[Abbott-Kane-Valiant’04,Bitansky-Paneth Rosen’15]

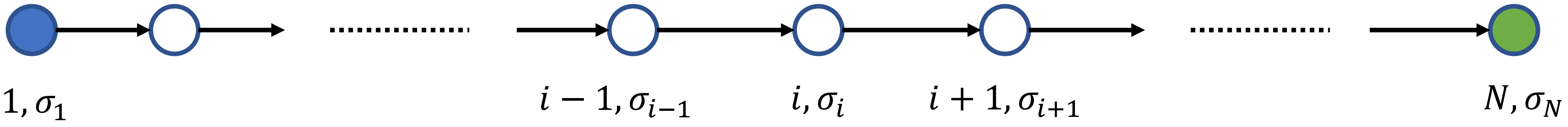


If the path is verifiable, then **Predecessor** is for free.

Based on reversible computation [Bennett’84]

Sink of Verifiable Line (SVL)

[Abbott-Kane-Valiant’04,Bitansky-Paneth Rosen’15]



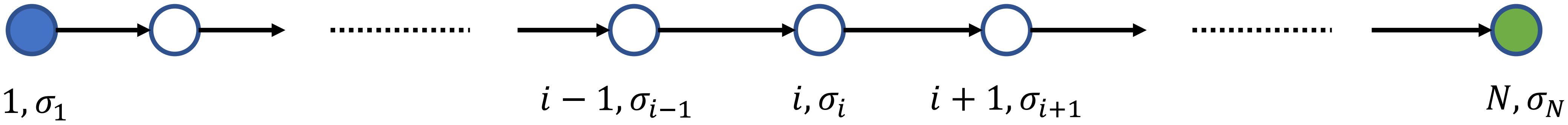
If the path is verifiable, then **Predecessor** is for free.

Based on reversible computation [Bennett’84]

[AKV’04,BPR’15]

Sink of Verifiable Line (SVL)

[Abbott-Kane-Valiant’04,Bitansky-Paneth Rosen’15]



If the path is verifiable, then **Predecessor** is for free.

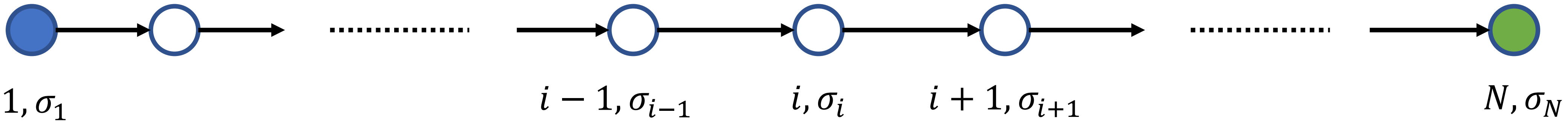
Based on reversible computation [Bennett’84]

iO hard → SVL hard → EOL hard

[BPR’15,GPS’16] [AKV’04,BPR’15]

Sink of Verifiable Line (SVL)

[Abbott-Kane-Valiant’04,Bitansky-Paneth Rosen’15]

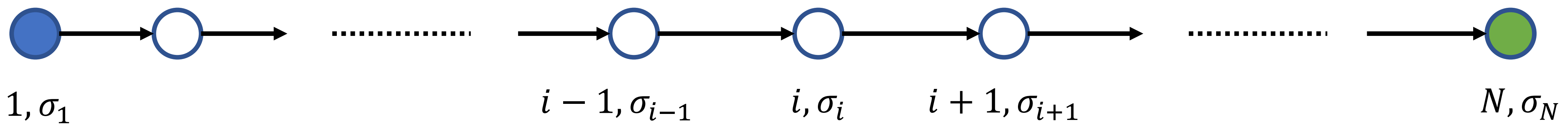


If the path is verifiable, then **Predecessor** is for free.

Based on reversible computation [Bennett’84]

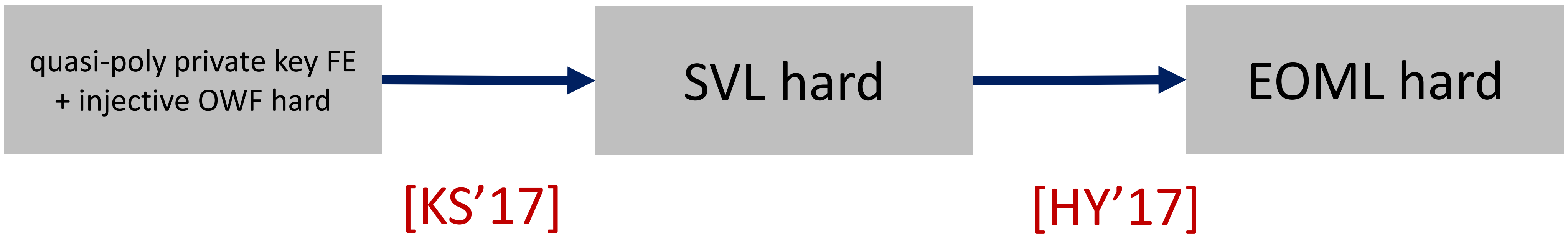
Sink of Verifiable Line (SVL)

[Abbott-Kane-Valiant’04,Bitansky-Paneth Rosen’15]



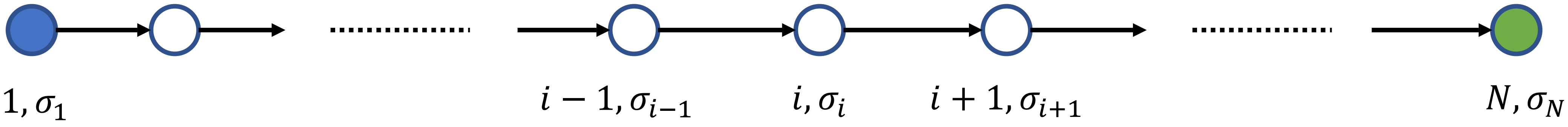
If the path is verifiable, then **Predecessor** is for free.

Based on reversible computation [Bennett’84]



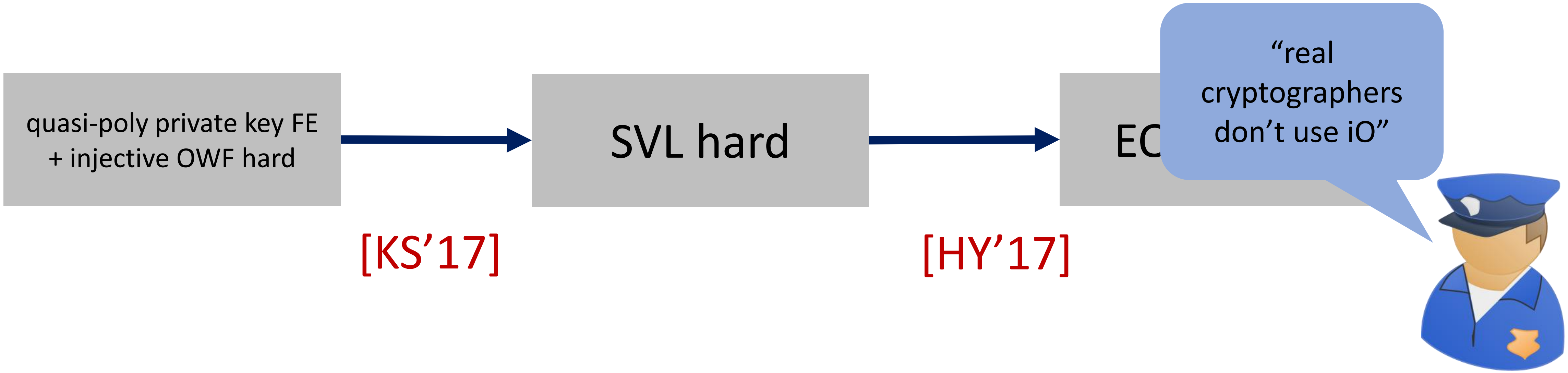
Sink of Verifiable Line (SVL)

[Abbott-Kane-Valiant'04,Bitansky-Paneth Rosen'15]



If the path is verifiable, then **Predecessor** is for free.

Based on reversible computation [Bennett'84]



Our Result

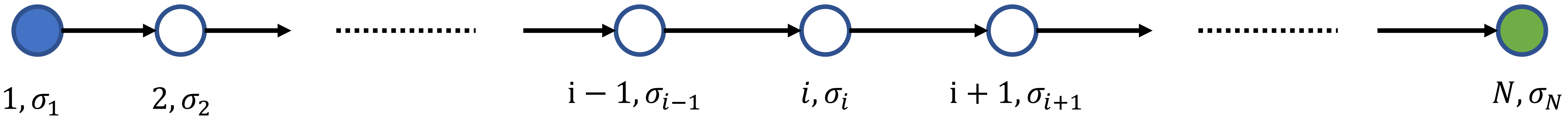
Our Result

CLS is as hard as breaking soundness of **Fiat-Shamir** when applied to the **sumcheck** protocol

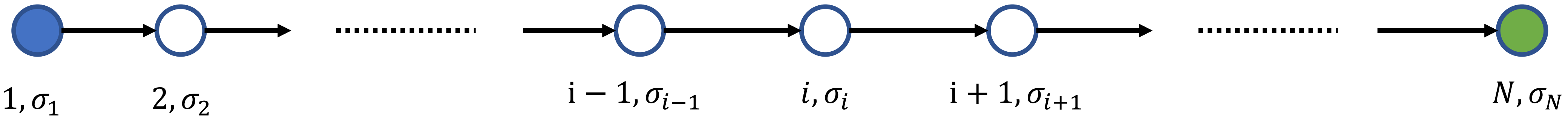
Our Construction

SVL Is No Easier Than Breaking Fiat-Shamir

Basic Idea

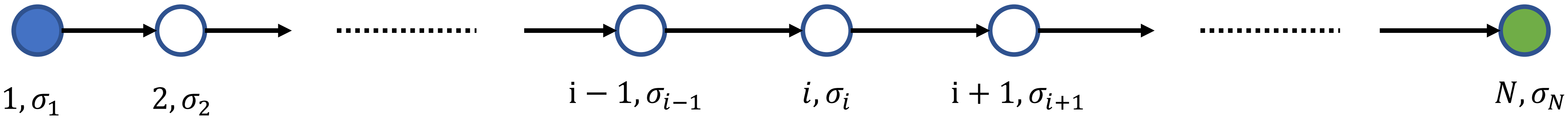

$1, \sigma_1$
$2, \sigma_2$
$\mathrm{i}-1, \sigma_{i-1}$
i, σ_i
$\mathrm{i}+1, \sigma_{i+1}$
N, σ_N

Basic Idea

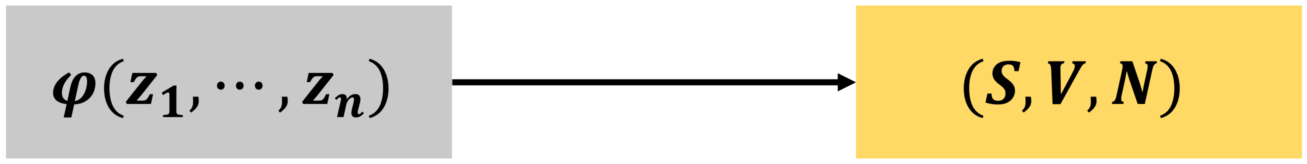


Reduce to SVL from #SAT

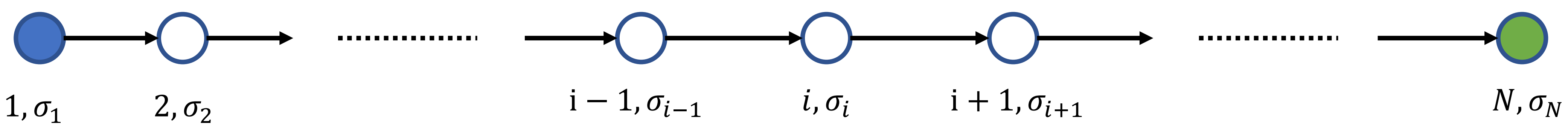
Basic Idea



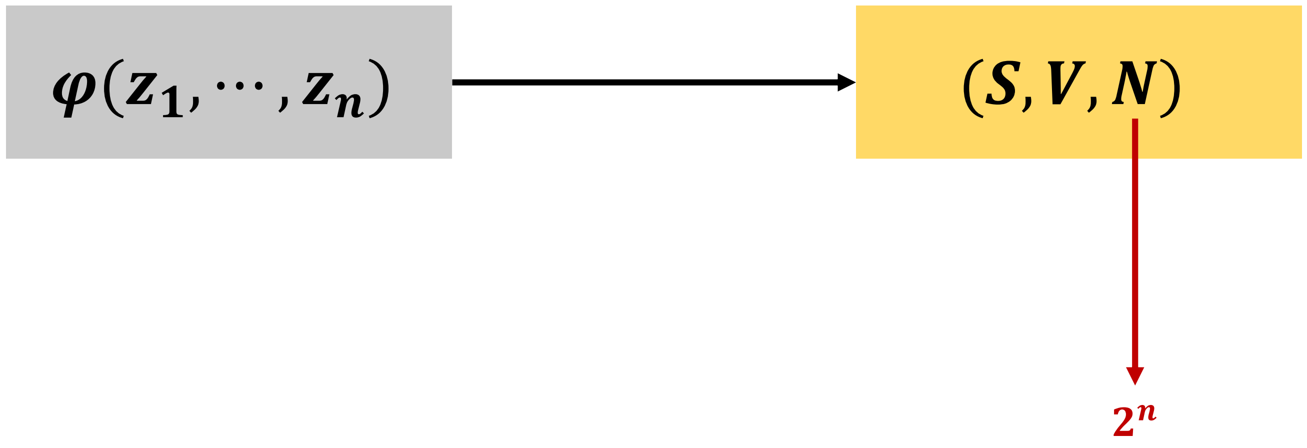
Reduce to SVL from #SAT



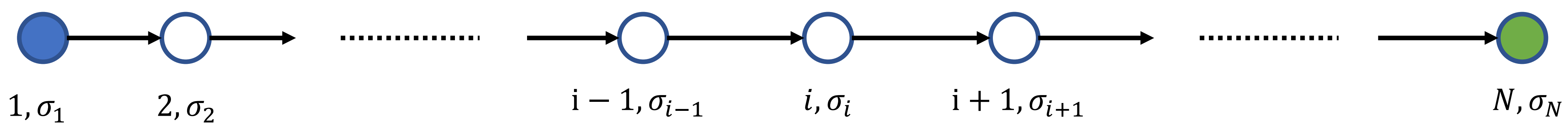
Basic Idea



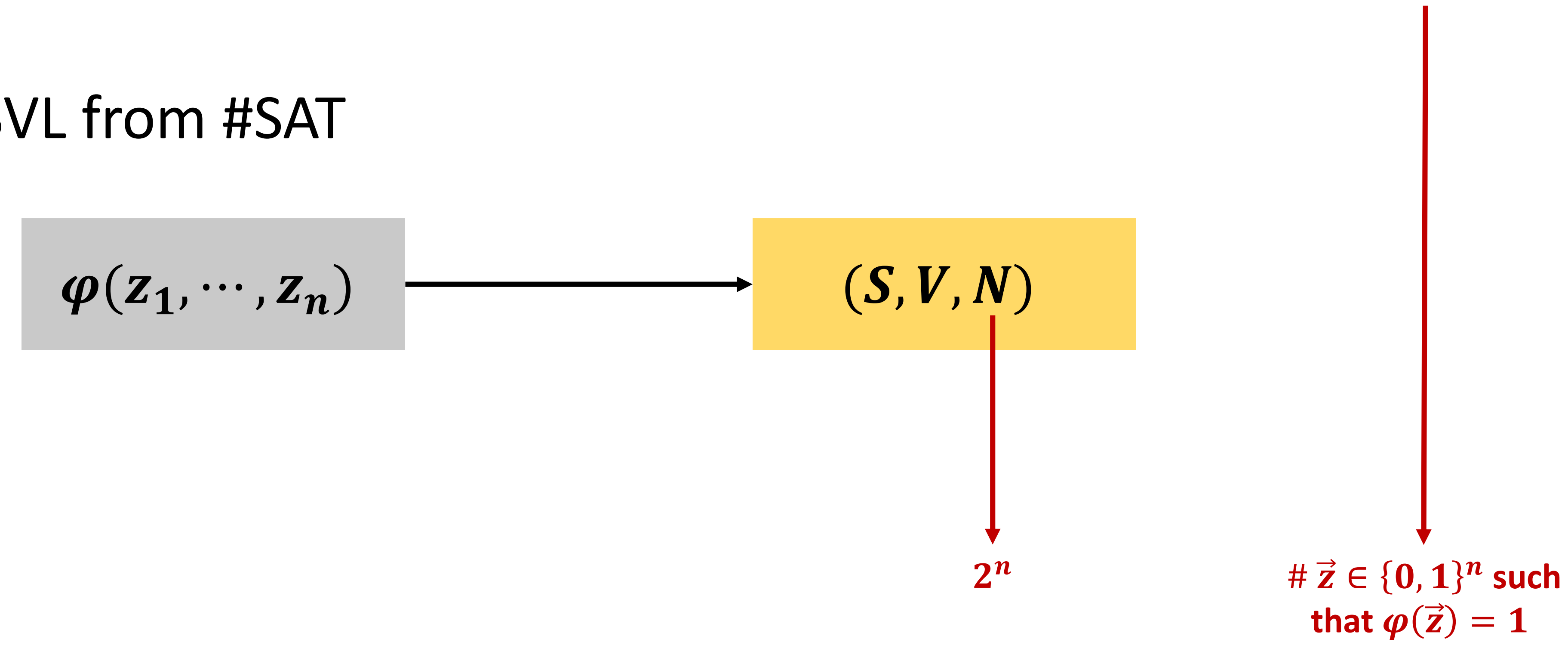
Reduce to SVL from #SAT



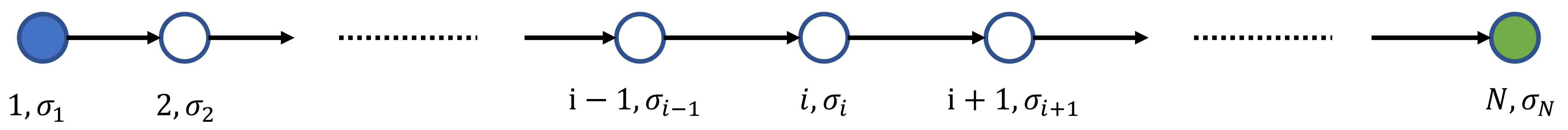
Basic Idea



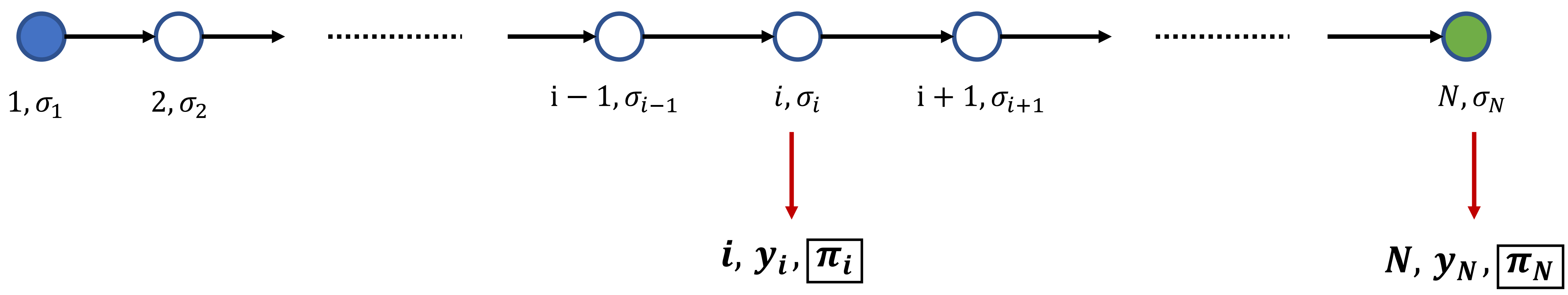
Reduce to SVL from #SAT



Basic Idea

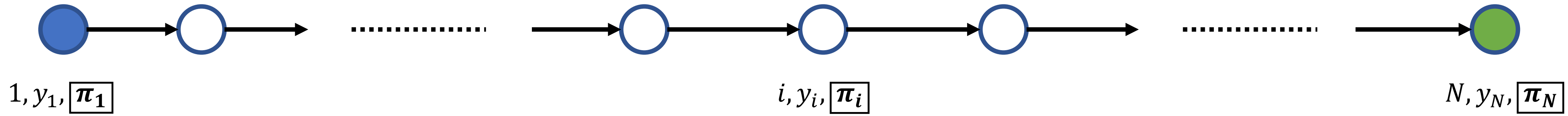

$1, \sigma_1$
$2, \sigma_2$
$\mathrm{i}-1, \sigma_{i-1}$
i, σ_i
$\mathrm{i}+1, \sigma_{i+1}$
N, σ_N

Basic Idea

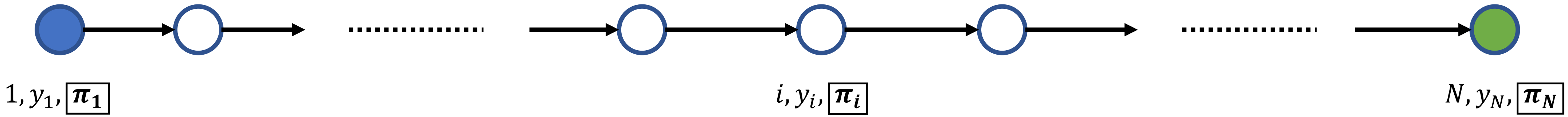


$1, \sigma_1$
$2, \sigma_2$
$\mathrm{i} - 1, \sigma_{i-1}$
i, σ_i
$\mathrm{i} + 1, \sigma_{i+1}$
N, σ_N
$\boldsymbol{i}, \boldsymbol{y}_{\boldsymbol{i}}, \boxed{\boldsymbol{\pi}_{\boldsymbol{i}}}$
$\boldsymbol{N}, \boldsymbol{y}_{\boldsymbol{N}}, \boxed{\boldsymbol{\pi}_{\boldsymbol{N}}}$

Basic Idea

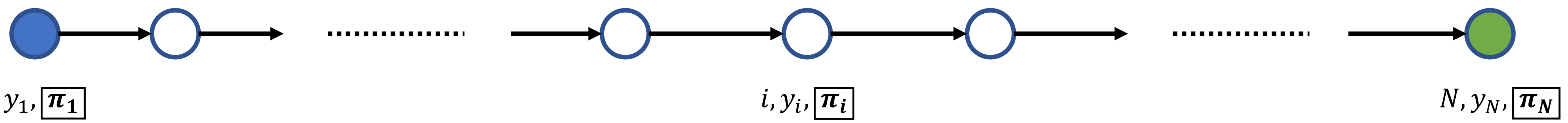

$1, y_1, \boxed{\boldsymbol{\pi_1}}$
$i, y_i, \boxed{\boldsymbol{\pi_i}}$
$N, y_N, \boxed{\boldsymbol{\pi_N}}$

Basic Idea



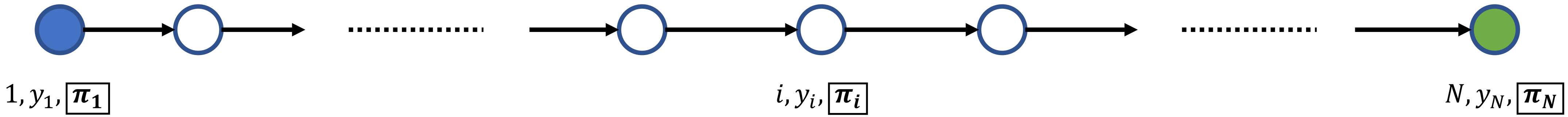
$\boldsymbol{V}\left(\boldsymbol{i},\, \boldsymbol{y_i}, \boxed{\boldsymbol{\pi_i}}\right)$ = ACCEPT $\iff$ y_i is the # of $\vec{z} \leq i$ such that $\varphi(\vec{z}) = 1$

Basic Idea

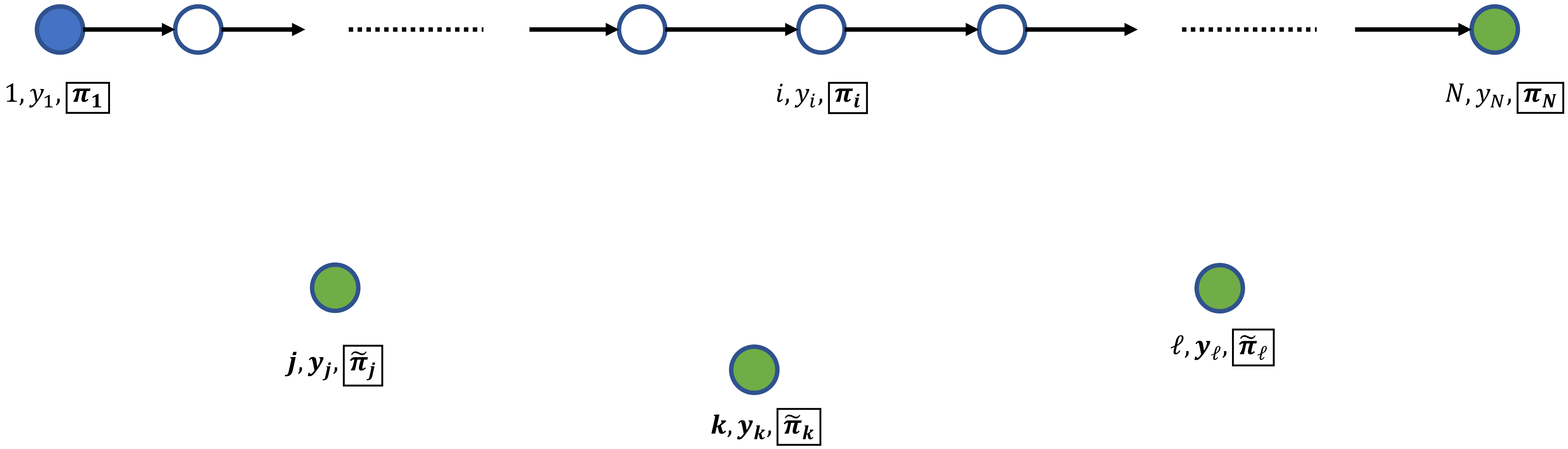


$$\boldsymbol{S\left(i,\ y_i, \boxed{\pi_i}\right) = i + 1,\ y_{i+1}, \boxed{\pi_{i+1}}}$$

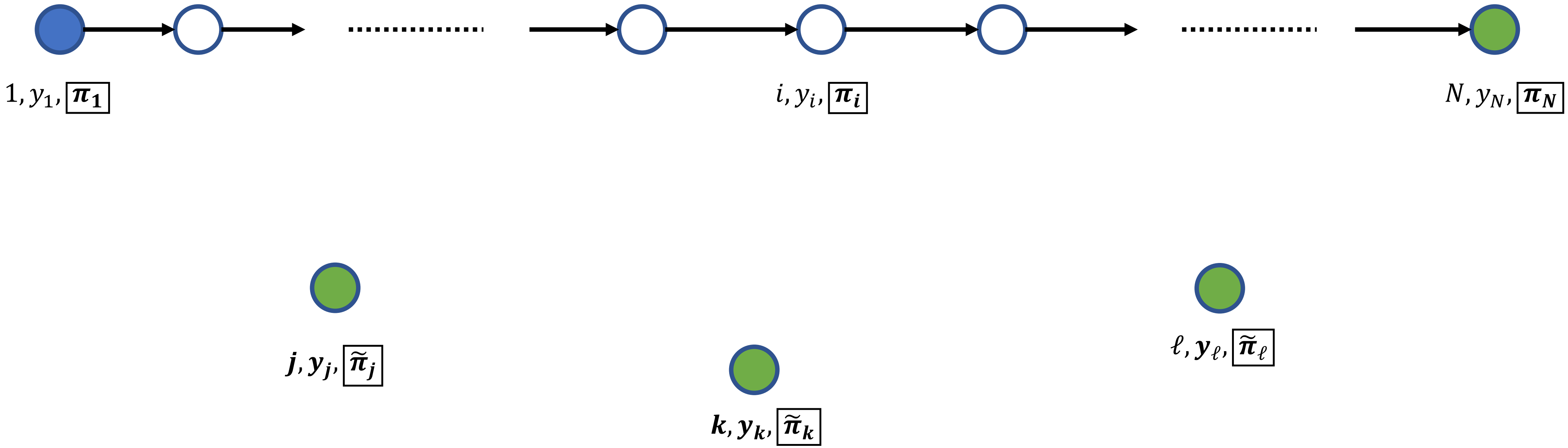
Relaxed SVL (rSVL)



Relaxed SVL (rSVL)



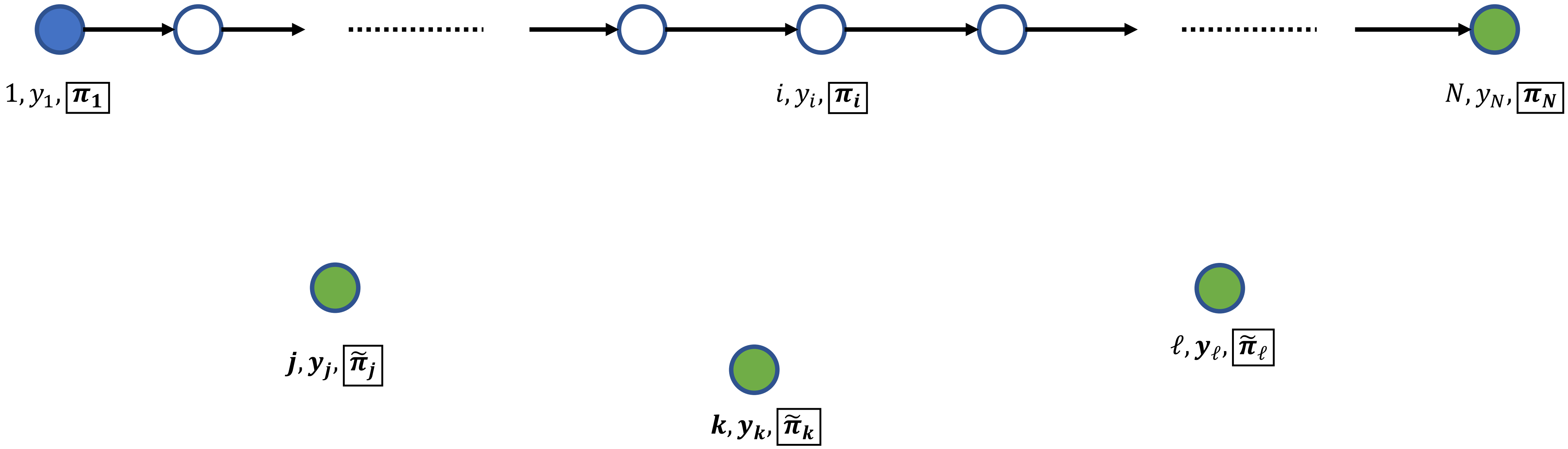
Relaxed SVL (rSVL)



Should be hard to find ``off-path'' $\boldsymbol{j}, \boldsymbol{y}_{\boldsymbol{j}}, \boxed{\widetilde{\boldsymbol{\pi}}_{\boldsymbol{j}}}$ such that

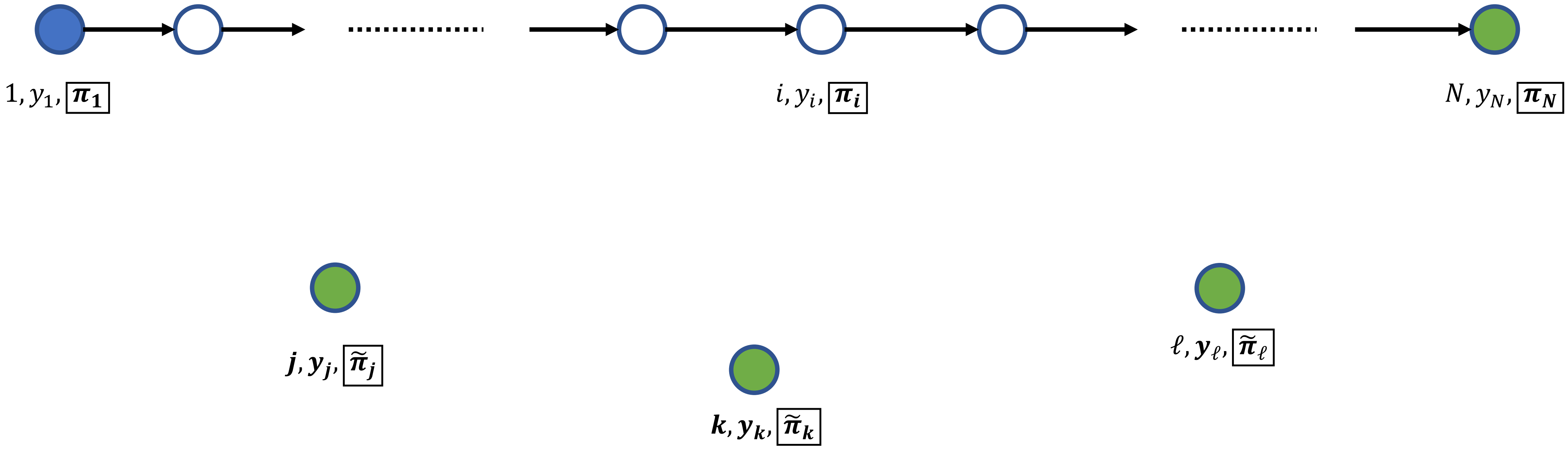
$$\boldsymbol{V}\left(\boldsymbol{j}, \boldsymbol{y}_{\boldsymbol{j}}, \boxed{\widetilde{\boldsymbol{\pi}}_{\boldsymbol{j}}}\right) = \texttt{ACCEPT}$$

Relaxed SVL (rSVL)



Solving an instance of rSVL

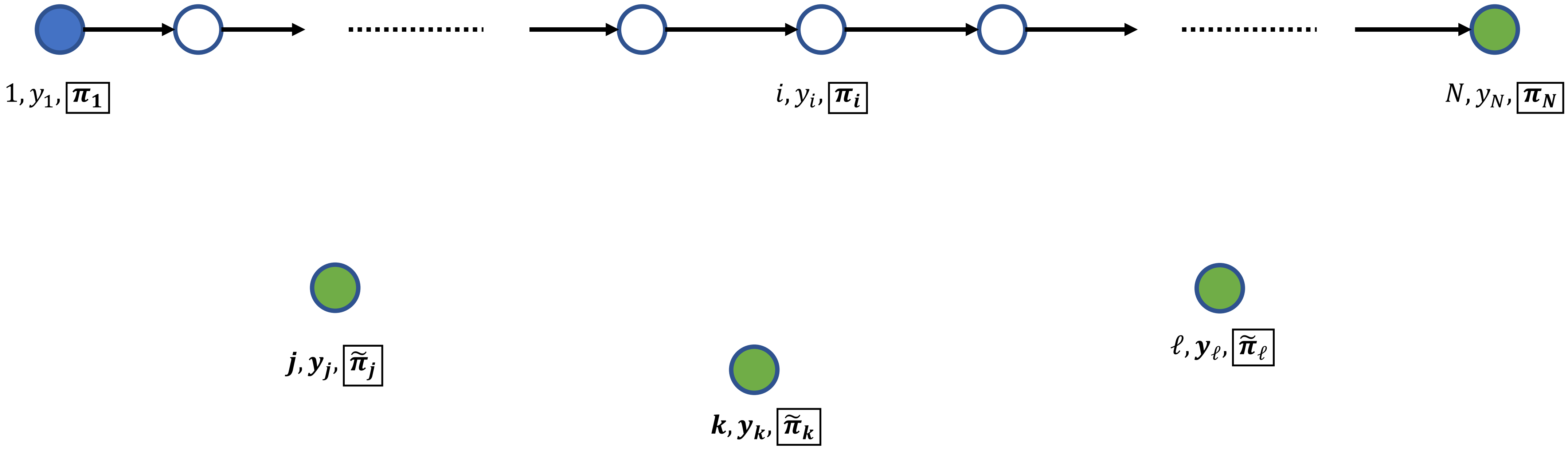
Relaxed SVL (rSVL)



Solving an instance of rSVL

solve #SAT instance φ

Relaxed SVL (rSVL)

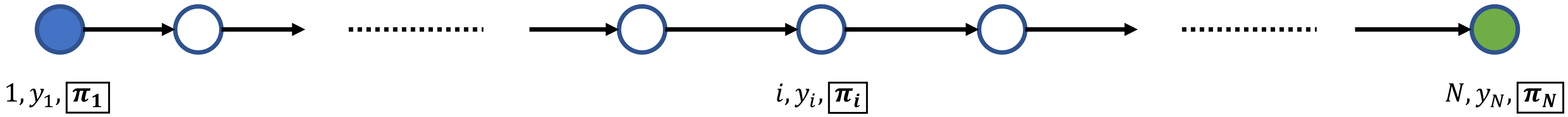


Solving an instance of rSVL

solve #SAT instance φ

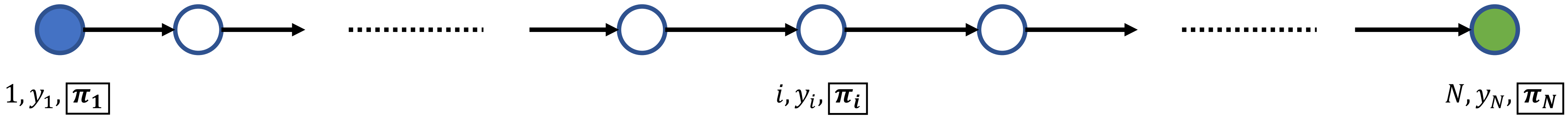
break (computational) soundness of $\boxed{\boldsymbol{\pi}}$

Challenges



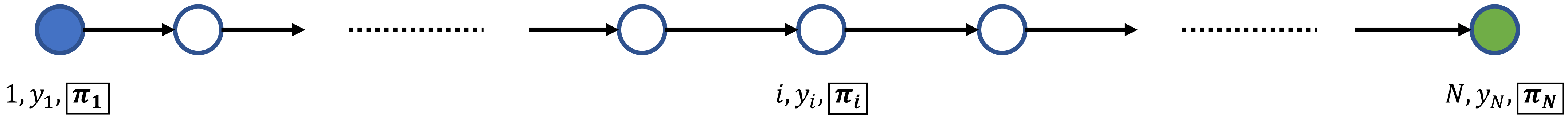
$1, y_1, \boxed{\boldsymbol{\pi_1}}$
$i, y_i, \boxed{\boldsymbol{\pi_i}}$
$N, y_N, \boxed{\boldsymbol{\pi_N}}$

Challenges



Several Challenges:

Challenges



Several Challenges:

Proof size has to be polynomial

Sumcheck Protocol [Lund-Fortnow-Karloff-Nisan'90]

Sumcheck Protocol [Lund-Fortnow-Karloff-Nisan'90]

Sumcheck Protocol [Lund-Fortnow-Karloff-Nisan’90]

Arithmetization

Sumcheck Protocol [Lund-Fortnow-Karloff-Nisan'90]

Arithmetization

$$\boldsymbol{\varphi}(\boldsymbol{z_1}, \cdots, \boldsymbol{z_n}) \longrightarrow \boldsymbol{f}\colon \mathbb{F}^{\boldsymbol{n}} \to \mathbb{F}$$

Sumcheck Protocol [Lund-Fortnow-Karloff-Nisan'90]

Arithmetization

$$\boldsymbol{\varphi}(\boldsymbol{z}_1, \cdots, \boldsymbol{z}_n) \longrightarrow \boldsymbol{f}\colon \mathbb{F}^{\boldsymbol{n}} \to \mathbb{F}$$

$$\text{degree}(f) = d = 4$$

Sumcheck Protocol [Lund-Fortnow-Karloff-Nisan'90]

Arithmetization

$$\boldsymbol{\varphi(z_1, \cdots, z_n)} \longrightarrow \boldsymbol{f} \colon \mathbb{F}^{\boldsymbol{n}} \to \mathbb{F}$$

$\text{degree}(f) = d = 4$

#SAT $\leq$ #3SAT4

Sumcheck Protocol [Lund-Fortnow-Karloff-Nisan’90]

Arithmetization

$$\boldsymbol{\varphi(z_1, \cdots, z_n)} \longrightarrow \boldsymbol{f} \colon \mathbb{F}^{\boldsymbol{n}} \to \mathbb{F}$$

$\text{degree}(f) = d = 4$

#SAT $\leq$ #3SAT4

Number of $\vec{z} \in \{0,1\}^n$ such that $\varphi(\vec{z}) = 1$ is

$$y = \sum_{\vec{z} \in \{0,1\}} f(\vec{z})$$

Sumcheck Protocol

Sumcheck

The sum $\sum_{z\in\{0,1\}^n} f(z)$ is some value y

Sumcheck

The sum $\sum_{z\in\{0,1\}^n} f(z)$ is some value y

Prove it!

Sumcheck

The sum $\sum_{z \in \{0,1\}^n} f(z)$ is some value y

Prove it!

$$\tilde{g}_1(x) \coloneqq \sum_{z_2, \cdots, z_n \in \{0,1\}} f(x, z_2, \cdots, z_n)$$

Sumcheck

The sum $\sum_{z\in\{0,1\}^n} f(z)$ is some value y

Prove it!

$$\tilde{g}_1(x) \coloneqq \sum_{z_2,\cdots,z_n\in\{0,1\}} f(x, z_2, \cdots, z_n) \xrightarrow{\tilde{g}_1(x)}$$

Sumcheck

The sum $\sum_{z \in \{0,1\}^n} f(z)$ is some value y

Prove it!

$$\tilde{g}_1(x) := \sum_{z_2, \cdots, z_n \in \{0,1\}} f(x, z_2, \cdots, z_n) \xrightarrow{\tilde{g}_1(x)}$$

$$\tilde{g}_1(0) + \tilde{g}_1(1) \stackrel{?}{=} y$$

Sumcheck

The sum $\sum_{z\in\{0,1\}^n} f(z)$ is some value y

Prove it!

$$\tilde{g}_1(x) \coloneqq \sum_{z_2,\cdots,z_n\in\{0,1\}} f(x, z_2, \cdots, z_n) \xrightarrow{\tilde{g}_1(x)}$$

$\{\tilde{g}_1(0), \cdots, \tilde{g}_1(d)\}$

$$\tilde{g}_1(0) + \tilde{g}_1(1) \stackrel{?}{=} y$$

Sumcheck

The sum $\sum_{z\in\{0,1\}^n} f(z)$ is some value y

Prove it!

$$\tilde{g}_1(x) \coloneqq \sum_{z_2,\cdots,z_n\in\{0,1\}} f(x, z_2, \cdots, z_n)$$

$$\xrightarrow[\{\tilde{g}_1(0),\cdots,\tilde{g}_1(d)\}]{\tilde{g}_1(x)}$$

$$\tilde{g}_1(0) + \tilde{g}_1(1) \stackrel{?}{=} y$$

$$\beta_1 \leftarrow_R \mathbb{F}$$

Sumcheck

The sum $\sum_{z\in\{0,1\}^n} f(z)$ is some value y

Prove it!

$$\tilde{g}_1(x) \coloneqq \sum_{z_2,\cdots,z_n\in\{0,1\}} f(x, z_2, \cdots, z_n)$$

$\tilde{g}_1(x)$

$\{\tilde{g}_1(0), \cdots, \tilde{g}_1(d)\}$

$$\tilde{g}_1(0) + \tilde{g}_1(1) \stackrel{?}{=} y$$

$$\beta_1 \leftarrow_R \mathbb{F}$$

β_1

Sumcheck

The sum $\sum_{z\in\{0,1\}^n} f(z)$ is some value y

Prove it!

$$\tilde{g}_1(x) \coloneqq \sum_{z_2,\cdots,z_n\in\{0,1\}} f(x, z_2, \cdots, z_n)$$

$\tilde{g}_1(x)$

$\{\tilde{g}_1(0), \cdots, \tilde{g}_1(d)\}$

$$\tilde{g}_1(0) + \tilde{g}_1(1) \stackrel{?}{=} y$$

$$\beta_1 \leftarrow_R \mathbb{F}$$

β_1

$$y_1 \coloneqq \tilde{g}_1(\beta_1)$$

$$y_1 \coloneqq \tilde{g}_1(\beta_1)$$

Sumcheck

The sum $\sum_{z\in\{0,1\}^n} f(z)$ is some value y

Prove it!

$$\tilde{g}_1(x) \coloneqq \sum_{z_2,\cdots,z_n\in\{0,1\}} f(x, z_2, \cdots, z_n)$$

$\tilde{g}_1(x)$

$\{\tilde{g}_1(0), \cdots, \tilde{g}_1(d)\}$

$$\tilde{g}_1(0) + \tilde{g}_1(1) \stackrel{?}{=} y$$

$$\beta_1 \leftarrow_R \mathbb{F}$$

β_1

$$y_1 \coloneqq \tilde{g}_1(\beta_1)$$

$$\tilde{g}_2(x) \coloneqq \sum_{z_3,\cdots,z_n\in\{0,1\}} f(\beta_1, x, z_3, \cdots, z_n)$$

$$y_1 \coloneqq \tilde{g}_1(\beta_1)$$

Sumcheck

The sum $\sum_{z\in\{0,1\}^n} f(z)$ is some value y

Prove it!

$$\tilde{g}_1(x) \coloneqq \sum_{z_2,\cdots,z_n\in\{0,1\}} f(x, z_2, \cdots, z_n)$$

$$\xrightarrow[\{\tilde{g}_1(0),\cdots,\tilde{g}_1(d)\}]{\tilde{g}_1(x)}$$

$$\tilde{g}_1(0) + \tilde{g}_1(1) \stackrel{?}{=} y$$

$$\beta_1 \leftarrow_R \mathbb{F}$$

$$\xleftarrow{\beta_1}$$

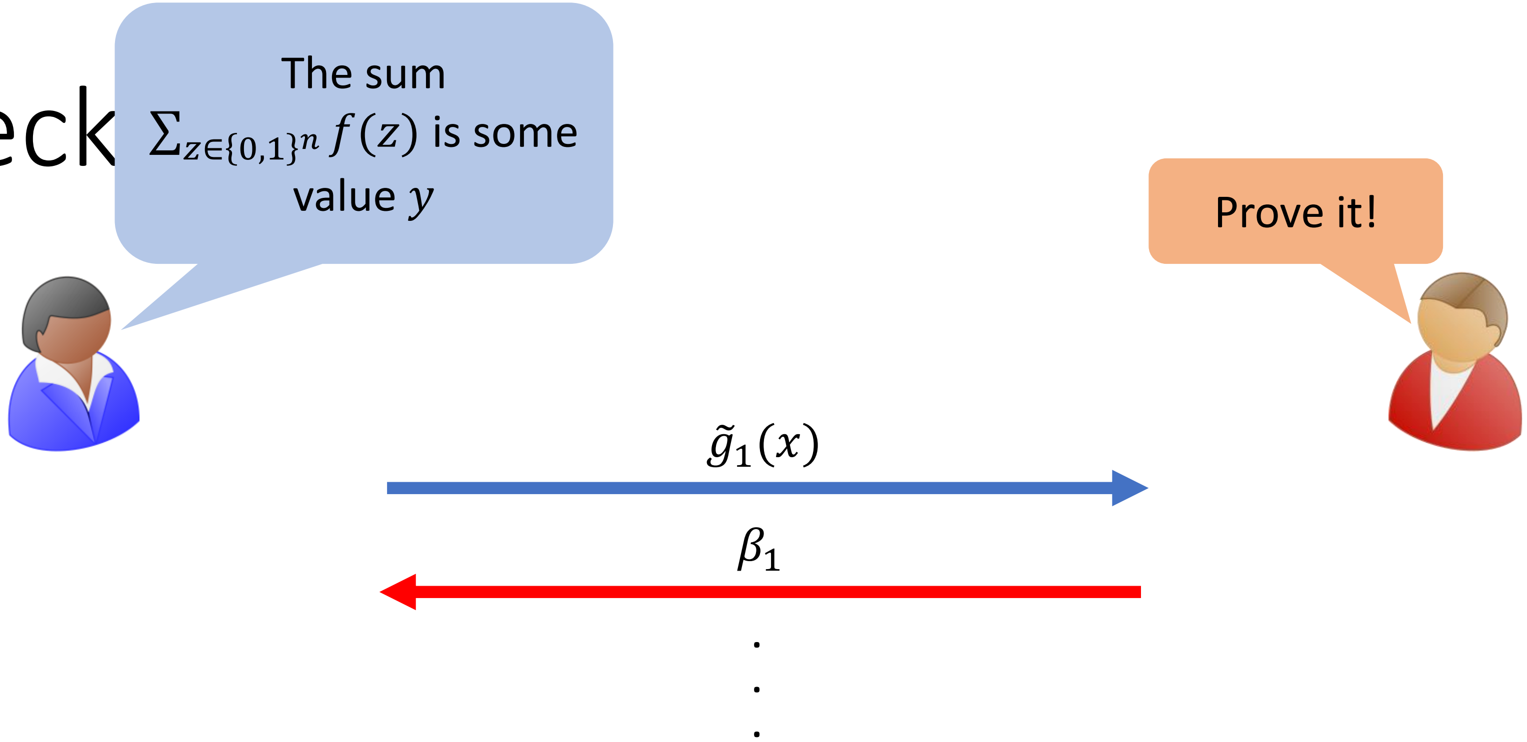
$$y_1 \coloneqq \tilde{g}_1(\beta_1)$$

$$\tilde{g}_2(x) \coloneqq \sum_{z_3,\cdots,z_n\in\{0,1\}} f(\beta_1, x, z_3, \cdots, z_n)$$

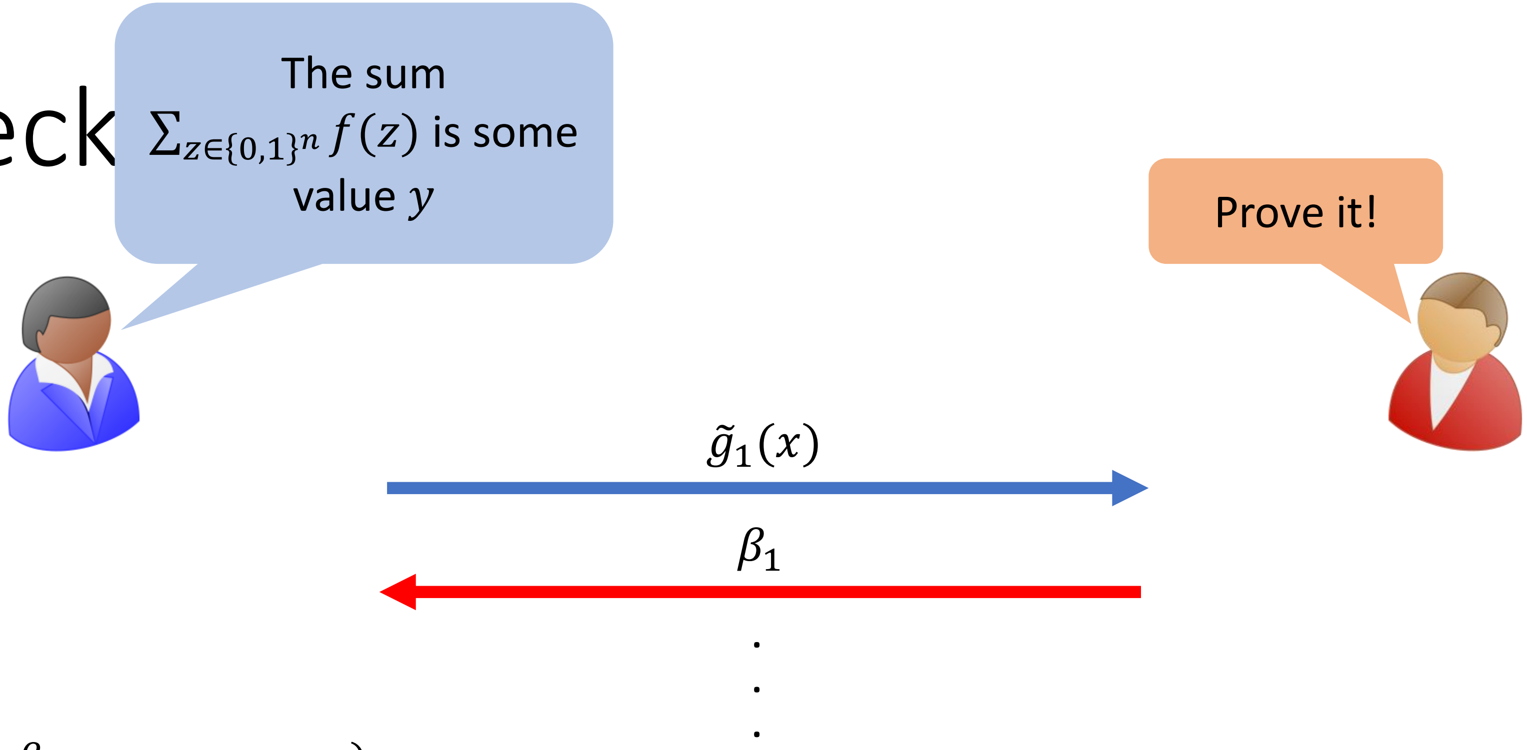
$$y_1 \coloneqq \tilde{g}_1(\beta_1)$$

$$\xrightarrow{\tilde{g}_2(x)}$$

Sumcheck


The sum $\Sigma_{z\in\{0,1\}^n} f(z)$ is some value y
Prove it!
$\tilde{g}_1(x)$
β_1
.
.
.

Sumcheck



$$\tilde{g}_j(x) \coloneqq \sum_{z_{j+1},\cdots,z_n\in\{0,1\}} f(\beta_1,\cdots,\beta_{j-1},x,z_{j+1},\cdots,z_n)$$

Sumcheck

The sum $\Sigma_{z\in\{0,1\}^n} f(z)$ is some value y

Prove it!

$\tilde{g}_1(x)$

β_1

$\vdots$

$\tilde{g}_j(x)$

$$\tilde{g}_j(x) \coloneqq \sum_{z_{j+1},\cdots,z_n\in\{0,1\}} f(\beta_1,\cdots,\beta_{j-1},x,z_{j+1},\cdots,z_n)$$

Sumcheck

The sum $\sum_{z\in\{0,1\}^n} f(z)$ is some value y

Prove it!

$\tilde{g}_1(x)$ →

← β_1

⋮

$\tilde{g}_j(x)$ →

$$\tilde{g}_j(x) \coloneqq \sum_{z_{j+1},\cdots,z_n\in\{0,1\}} f(\beta_1,\cdots,\beta_{j-1},x,z_{j+1},\cdots,z_n)$$

$$\tilde{g}_j(0)+\tilde{g}_j(1) \stackrel{?}{=} \tilde{g}_{j-1}(\beta_{j-1})$$

Sumcheck

The sum $\sum_{z \in \{0,1\}^n} f(z)$ is some value y

Prove it!

$\tilde{g}_1(x)$

β_1

$\vdots$

$\tilde{g}_j(x)$

$$\tilde{g}_j(x) \coloneqq \sum_{z_{j+1},\cdots,z_n \in \{0,1\}} f(\beta_1, \cdots, \beta_{j-1}, x, z_{j+1}, \cdots, z_n)$$

$$\tilde{g}_j(0) + \tilde{g}_j(1) \stackrel{?}{=} \tilde{g}_{j-1}(\beta_{j-1})$$

$$\beta_j \leftarrow_R \mathbb{F}$$

Sumcheck

The sum $\sum_{z \in \{0,1\}^n} f(z)$ is some value y

Prove it!

$\tilde{g}_1(x)$

β_1

$\vdots$

$$\tilde{g}_j(x) \coloneqq \sum_{z_{j+1},\cdots,z_n \in \{0,1\}} f(\beta_1, \cdots, \beta_{j-1}, x, z_{j+1}, \cdots, z_n)$$

$\tilde{g}_j(x)$

$$\tilde{g}_j(0) + \tilde{g}_j(1) \stackrel{?}{=} \tilde{g}_{j-1}(\beta_{j-1})$$

β_j

$$\beta_j \leftarrow_R \mathbb{F}$$

Sumcheck

The sum $\sum_{z\in\{0,1\}^n} f(z)$ is some value y

Prove it!

$\tilde{g}_1(x)$ →

← β_1

⋮

$$\tilde{g}_j(x) \coloneqq \sum_{z_{j+1},\cdots,z_n\in\{0,1\}} f(\beta_1,\cdots,\beta_{j-1},x,z_{j+1},\cdots,z_n)$$

$\tilde{g}_j(x)$ →

$$\tilde{g}_j(0)+\tilde{g}_j(1) \stackrel{?}{=} \tilde{g}_{j-1}(\beta_{j-1})$$

← β_j

$$\beta_j \leftarrow_R \mathbb{F}$$

$$y_j \coloneqq \tilde{g}_j(\beta_j)$$

***j*-th claim**

Sumcheck

The sum $\Sigma_{z\in\{0,1\}^n} f(z)$ is some value y

Prove it!

$\tilde{g}_1(x)$ →

← β_1

⋮

$$\tilde{g}_j(x) \coloneqq \sum_{z_{j+1},\cdots,z_n\in\{0,1\}} f(\beta_1,\cdots,\beta_{j-1},x,z_{j+1},\cdots,z_n)$$

$\tilde{g}_j(x)$ →

$$\tilde{g}_j(0)+\tilde{g}_j(1) \overset{?}{=} \tilde{g}_{j-1}(\beta_{j-1})$$

$$\beta_j \leftarrow_R \mathbb{F}$$

← β_j

$$y_j \coloneqq \tilde{g}_j(\beta_j)$$

***j*-th claim**

⋮

Sumcheck

The sum $\sum_{z\in\{0,1\}^n} f(z)$ is some value y

Prove it!

$\tilde{g}_1(x)$

β_1

$\vdots$

$$\tilde{g}_j(x) \coloneqq \sum_{z_{j+1},\cdots,z_n\in\{0,1\}} f(\beta_1,\cdots,\beta_{j-1},x,z_{j+1},\cdots,z_n)$$

$\tilde{g}_j(x)$

$$\tilde{g}_j(0)+\tilde{g}_j(1)\stackrel{?}{=}\tilde{g}_{j-1}(\beta_{j-1})$$

β_j

$\beta_j \leftarrow_R \mathbb{F}$

$y_j \coloneqq \tilde{g}_j(\beta_j)$

***j*-th claim**

$\vdots$

β_{n-1}

$\tilde{g}_n(x)$

Sumcheck

The sum $\sum_{z\in\{0,1\}^n} f(z)$ is some value y

Prove it!

$\tilde{g}_1(x)$

β_1

$\vdots$

$$\tilde{g}_j(x) := \sum_{z_{j+1},\cdots,z_n\in\{0,1\}} f(\beta_1,\cdots,\beta_{j-1},x,z_{j+1},\cdots,z_n)$$

$\tilde{g}_j(x)$

$$\tilde{g}_j(0)+\tilde{g}_j(1) \stackrel{?}{=} \tilde{g}_{j-1}(\beta_{j-1})$$

β_j

$\beta_j \leftarrow_R \mathbb{F}$

$y_j := \tilde{g}_j(\beta_j)$

***j*-th claim**

$\vdots$

β_{n-1}

$\beta_n \leftarrow_R \mathbb{F}$

$\tilde{g}_n(x)$

Sumcheck

The sum $\sum_{z\in\{0,1\}^n} f(z)$ is some value y

Prove it!

$\tilde{g}_1(x)$ →

← β_1

⋮

$$\tilde{g}_j(x) \coloneqq \sum_{z_{j+1},\cdots,z_n\in\{0,1\}} f(\beta_1,\cdots,\beta_{j-1}, x, z_{j+1},\cdots,z_n)$$

$\tilde{g}_j(x)$ →

$\tilde{g}_j(0)+\tilde{g}_j(1) \stackrel{?}{=} \tilde{g}_{j-1}(\beta_{j-1})$

← β_j

$\beta_j \leftarrow_R \mathbb{F}$

$y_j \coloneqq \tilde{g}_j(\beta_j)$

***j*-th claim**

⋮

← β_{n-1}

$\tilde{g}_n(x)$ →

$\beta_n \leftarrow_R \mathbb{F}$

$f(\beta_1,\cdots,\beta_n) \stackrel{?}{=} \tilde{g}_n(\beta_n)$

Sumcheck

The sum $\Sigma_{z\in\{0,1\}^n} f(z)$ is some value y

Prove it!

size $N = 2^n$
claim

$\tilde{g}_1(x)$

β_1

$\vdots$

$$\tilde{g}_j(x) \coloneqq \sum_{z_{j+1},\cdots,z_n\in\{0,1\}} f(\beta_1,\cdots,\beta_{j-1},x,z_{j+1},\cdots,z_n)$$

$\tilde{g}_j(x)$

$$\tilde{g}_j(0)+\tilde{g}_j(1) \stackrel{?}{=} \tilde{g}_{j-1}(\beta_{j-1})$$

β_j

$$y_j \coloneqq \tilde{g}_j(\beta_j)$$

j-th claim

$\beta_j \leftarrow_R \mathbb{F}$

$\vdots$

β_{n-1}

$\tilde{g}_n(x)$

$\beta_n \leftarrow_R \mathbb{F}$

$$f(\beta_1,\cdots,\beta_n) \stackrel{?}{=} \tilde{g}_n(\beta_n)$$

Sumcheck

The sum $\sum_{z\in\{0,1\}^n} f(z)$ is some value y

Prove it!

size $N = 2^n$ claim

$\tilde{g}_1(x)$ →

← β_1

⋮

$$\tilde{g}_j(x) \coloneqq \sum_{z_{j+1},\cdots,z_n\in\{0,1\}} f(\beta_1,\cdots,\beta_{j-1},x,z_{j+1},\cdots,z_n)$$

$\tilde{g}_j(x)$ →

$\tilde{g}_j(0) + \tilde{g}_j(1) \stackrel{?}{=} \tilde{g}_{j-1}(\beta_{j-1})$

← β_j

$\beta_j \leftarrow_R \mathbb{F}$

$y_j \coloneqq \tilde{g}_j(\beta_j)$

***j*-th claim**

⋮

← β_{n-1}

size $N/2^j$ claim

$\tilde{g}_n(x)$ →

$\beta_n \leftarrow_R \mathbb{F}$

$f(\beta_1,\cdots,\beta_n) \stackrel{?}{=} \tilde{g}_n(\beta_n)$

The j-th claim

$$y_j \coloneqq \tilde{g}_j(\beta_j)$$

The j-th claim

$$y_j := \tilde{g}_j(\beta_j)$$

$$\sum_{z_{j+1},\cdots,z_n \in \{0,1\}} f(\beta_1,\cdots,\beta_{j-1},\beta_j,z_{j+1},\cdots,z_n) \stackrel{?}{=} y_j$$

The j-th claim

$$y_j := \tilde{g}_j(\beta_j)$$

$$\sum_{z_{j+1},\cdots,z_n \in \{0,1\}} f(\beta_1, \cdots, \beta_{j-1}, \beta_j, z_{j+1}, \cdots, z_n) \stackrel{?}{=} y_j$$

size $N/2^j$ claim

The j-th claim

$$y_j := \tilde{g}_j(\beta_j)$$

$$\sum_{z_{j+1},\cdots,z_n\in\{0,1\}} f(\beta_1,\cdots,\beta_{j-1},\beta_j,z_{j+1},\cdots,z_n) \stackrel{?}{=} y_j$$

size $N/2^j$ claim

Recall

$$\tilde{g}_j(x) := \sum_{z_{j+1},\cdots,z_n\in\{0,1\}} f(\beta_1,\cdots,\beta_{j-1},x,z_{j+1},\cdots,z_n)$$

Soundness

Soundness: if the j-th claim is **false** then $\forall \tilde{g}_{j+1}(x)$ the $(j+1)$-th claim is **also false**

Soundness

Soundness: if the j-th claim is **false** then $\forall \tilde{g}_{j+1}(x)$ the $(j+1)$-th claim is **also false**

Unambiguous Soundness: if $\tilde{g}_{j+1}(x) \neq g_{j+1}(x)$, then the $(j+1)$-th claim is **false** **even if $\boldsymbol{j}$-th claim was true**

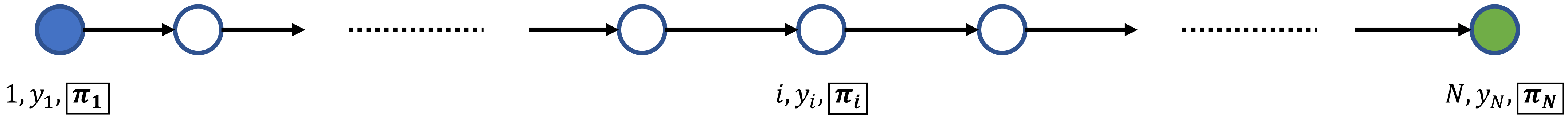
Soundness

Soundness: if the j-th claim is **false** then $\forall \tilde{g}_{j+1}(x)$ the $(j+1)$-th claim is **also false**

Unambiguous Soundness: if $\tilde{g}_{j+1}(x) \neq g_{j+1}(x)$, then the $(j+1)$-th claim is **false** **even if *j*-th claim was true**

Both with high probability over β_{j+1} - Schwartz-Zippel.

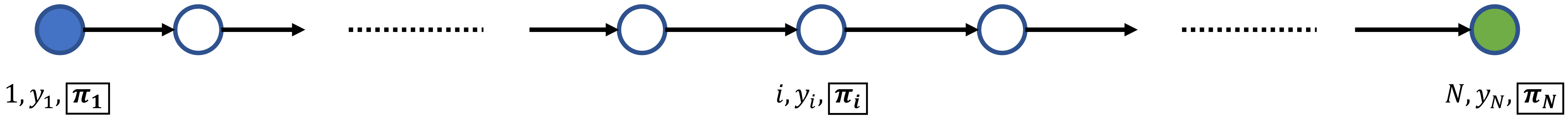
Basic Idea



Several Challenges:

Proof size has to be polynomial

Basic Idea

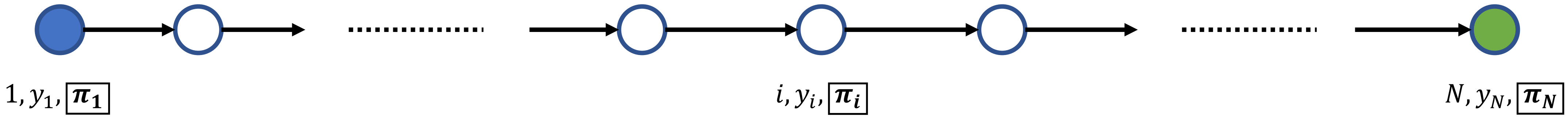


Several Challenges:

Proof size has to be polynomial

Sumcheck protocol

Basic Idea



Several Challenges:

Proof size has to be polynomial

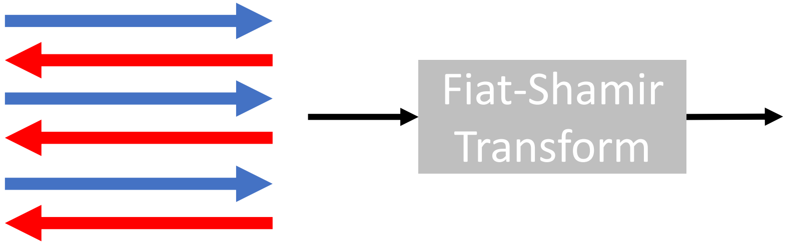
Sumcheck protocol

Sumcheck Protocol is interactive

[Fiat-Shamir'86] Transformation

[Fiat-Shamir'86] Transformation


[Fiat-Shamir'86] Transformation



[Fiat-Shamir'86] Transformation

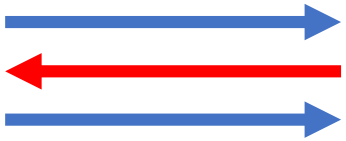
[Fiat-Shamir'86] Transformation

Replaced by a hash function h

[Fiat-Shamir'86] Transformation

Replaced by a hash function h

$=h($



$)$

Fiat-Shamir for Sumcheck

$$\xleftarrow{\quad\beta_j\quad} \;=\; h\left(\begin{array}{c} \xrightarrow{\quad\tilde{g}_1(x)\quad} \\ \xleftarrow{\quad\beta_1\quad} \\ \vdots \\ \xrightarrow{\quad\tilde{g}_j(x)\quad} \end{array} \right)$$

Fiat-Shamir for Sumcheck

Assumption

Resulting non-interactive (determinstic) argument is (adaptively) unambiguously sound

Fiat-Shamir for Sumcheck

Assumption

Resulting non-interactive (determinstic) argument is (adaptively) unambiguously sound

Given h, no poly-time prover can find accepting proof:

Fiat-Shamir for Sumcheck

Assumption

Resulting non-interactive (determinstic) argument is (adaptively) unambiguously sound

Given h, no poly-time prover can find accepting proof:

1. $\boxed{\pi}$ for a false statement y, or

Fiat-Shamir for Sumcheck

Assumption

Resulting non-interactive (determinstic) argument is (adaptively) unambiguously sound

Given h, no poly-time prover can find accepting proof:

1. $\boxed{\pi}$ for a false statement y, or
2. $\boxed{\tilde{\pi}} \neq \boxed{\pi}$ for true statement y

Fiat-Shamir for Sumcheck

Assumption

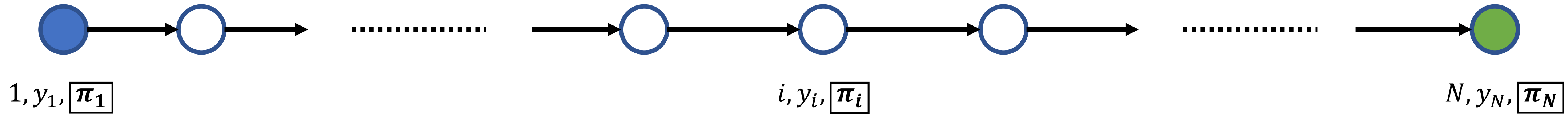
Resulting non-interactive (determinstic) argument is (adaptively) unambiguously sound

Given h, no poly-time prover can find accepting proof:

1. $\boxed{\pi}$ for a false statement y, or
2. $\boxed{\tilde{\pi}} \neq \boxed{\pi}$ for true statement y

True if h is a random oracle.

Basic Idea



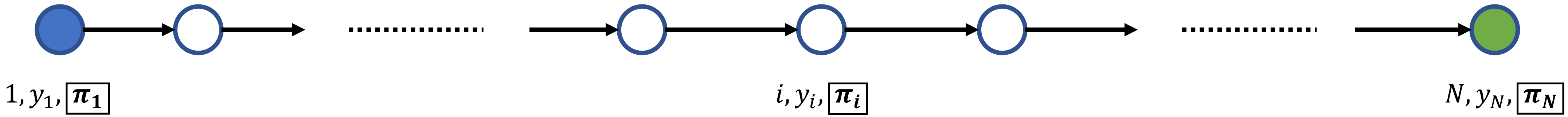
Several Challenges:

Proof size has to be polynomial

Sumcheck protocol

Sumcheck Protocol is interactive

Basic Idea



Several Challenges:

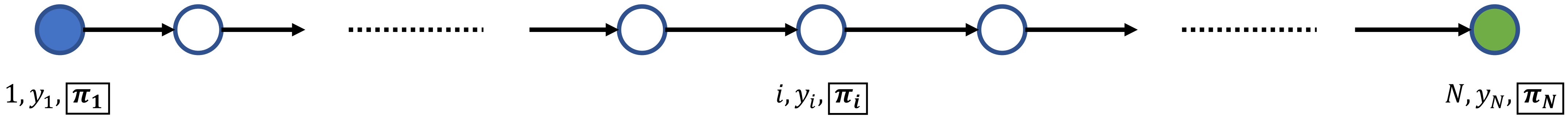
Proof size has to be polynomial

Sumcheck protocol

Sumcheck Protocol is interactive

Fiat-Shamir Transform

Basic Idea



Several Challenges:

Proof size has to be polynomial

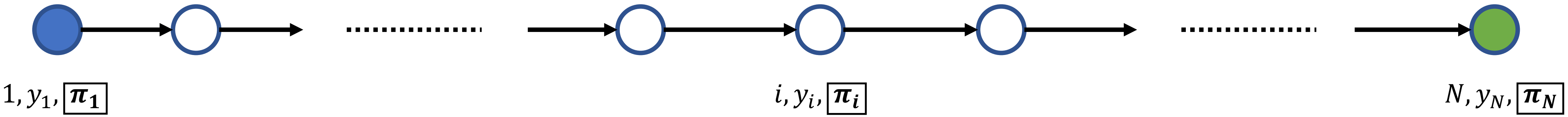
Sumcheck protocol

Sumcheck Protocol is interactive

Fiat-Shamir Transform

Computing $\boldsymbol{S}\left(\boldsymbol{i}, \boldsymbol{y_i}, \boxed{\boldsymbol{\pi_i}}\right)$

Basic Idea



Several Challenges:

Proof size has to be polynomial

Sumcheck protocol

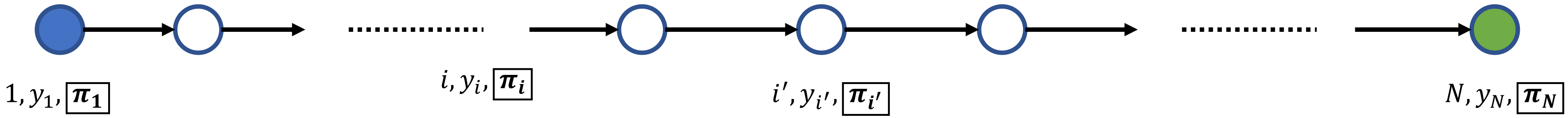
Sumcheck Protocol is interactive

Fiat-Shamir Transform

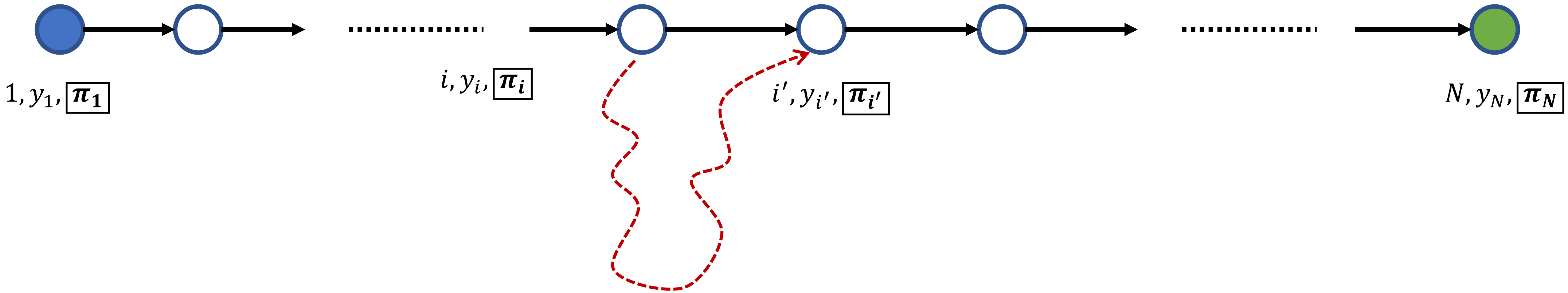
Computing $\boldsymbol{S}\left(\boldsymbol{i}, \boldsymbol{y_i}, \boxed{\boldsymbol{\pi_i}}\right)$

Incremental Proof Updates

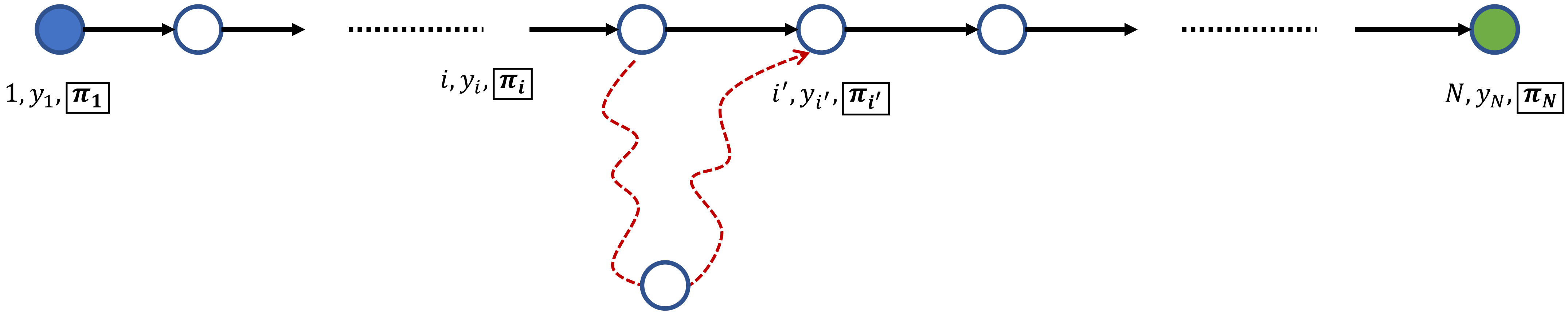
Incremental Proof Updates



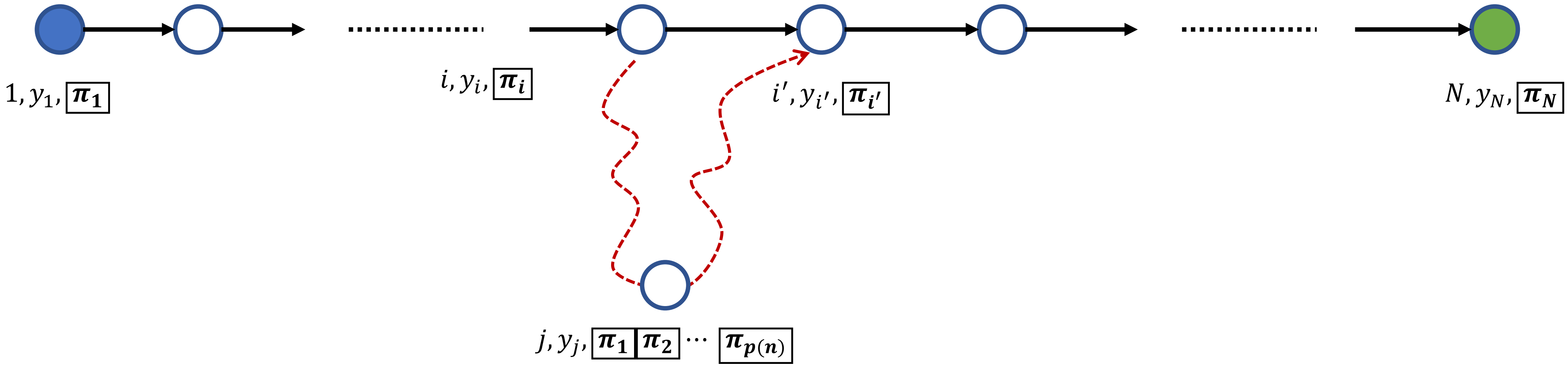
Incremental Proof Updates



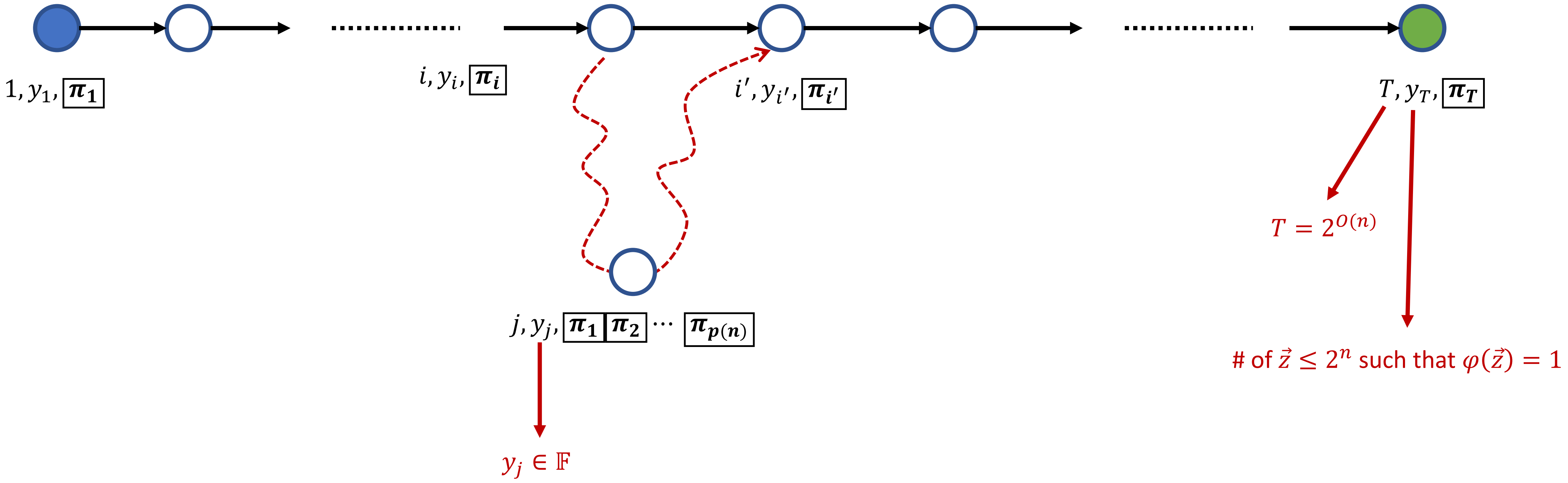
Incremental Proof Updates



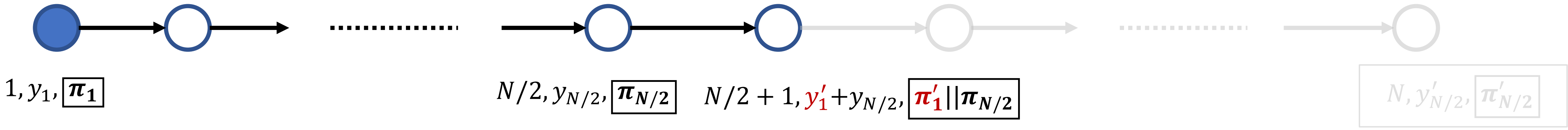
Incremental Proof Updates



Incremental Proof Updates



Naïve Recursive Construction

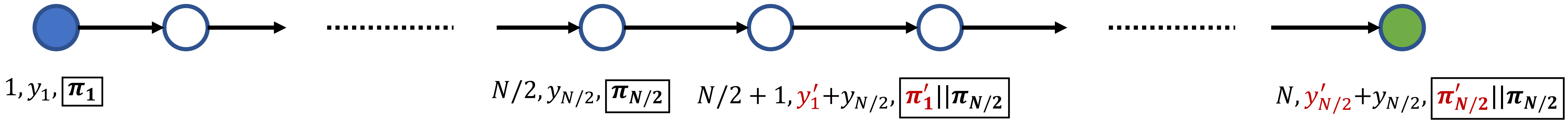


Construction for $N/2 \rightarrow$ to construction for N

First $N/2$ assignments: Do recursively

Second N/2 assignments: Add second proof $y_i, \boxed{\boldsymbol{\pi_i}}$

Naïve Recursive Construction

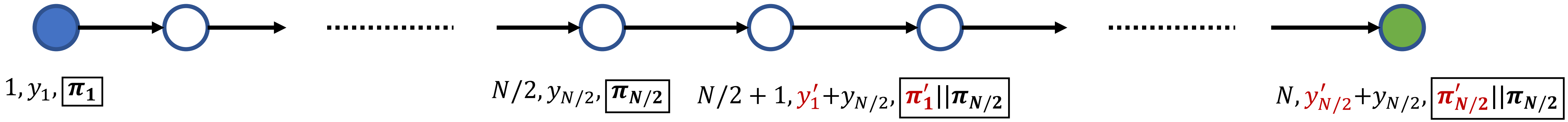


Construction for $N/2 \rightarrow$ to construction for N

First $N/2$ assignments: Do recursively

Second N/2 assignments: Add second proof $y_i, \boxed{\boldsymbol{\pi_i}}$

Naïve Recursive Construction



Construction for $N/2 \rightarrow$ to construction for N

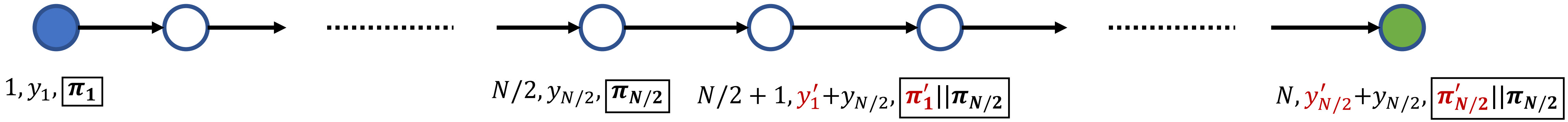
First $N/2$ assignments: Do recursively

Second N/2 assignments: Add second proof $y_i, \boxed{\boldsymbol{\pi_i}}$

Proof size: $\boldsymbol{P(N) = 2\ P(N/2)}$

 # Steps: $\boldsymbol{T(N) = 2\ T(N/2)}$

Naïve Recursive Construction



Construction for $N/2 \rightarrow$ to construction for N

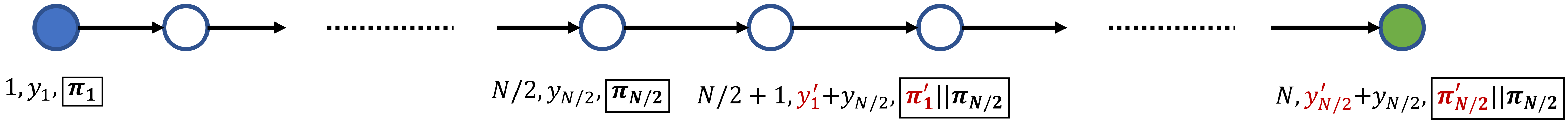
First $N/2$ assignments: Do recursively

Second N/2 assignments: Add second proof $y_i, \boxed{\pi_i}$

Proof size: $P(N) = 2\ P(N/2)$

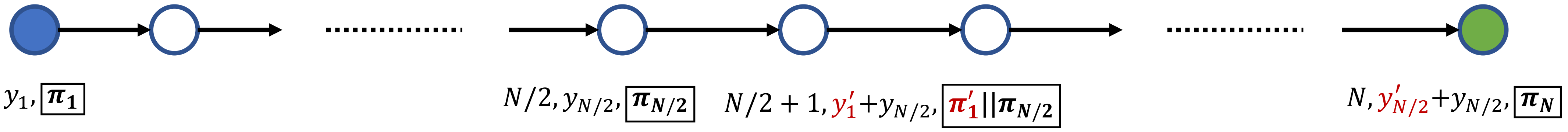
 # Steps: $T(N) = 2\ T(N/2)$

Merge Proofs [Valiant'06]



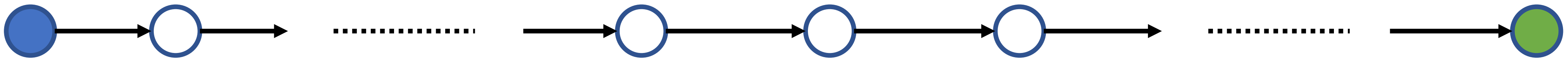
Merge proofs for $y_{N/2}, \boxed{\boldsymbol{\pi}_{N/2}}$ and $y'_{N/2}, \boxed{\boldsymbol{\pi}'_{N/2}}$

Merge Proofs [Valiant'06]



Merge proofs for $y_{N/2}, \boxed{\boldsymbol{\pi_{N/2}}}$ and $y'_{N/2}, \boxed{\boldsymbol{\pi'_{N/2}}}$

Merge Proofs [Valiant'06]



$1, y_1, \boxed{\boldsymbol{\pi}_1}$

$N/2, y_{N/2}, \boxed{\boldsymbol{\pi}_{N/2}}$

$N/2 + 1, y'_1 + y_{N/2}, \boxed{\boldsymbol{\pi}'_1 || \boldsymbol{\pi}_{N/2}}$

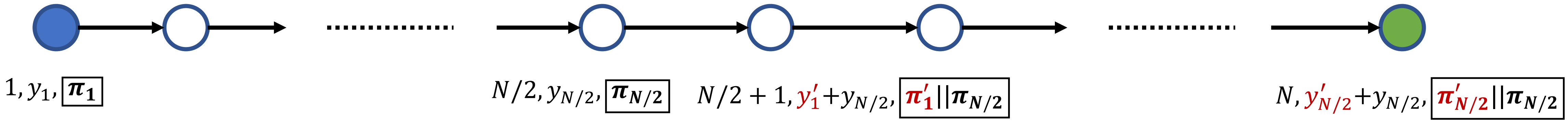
$N, y'_{N/2} + y_{N/2}, \boxed{\boldsymbol{\pi}'_{N/2} || \boldsymbol{\pi}_{N/2}}$

Merge proofs for $y_{N/2}, \boxed{\boldsymbol{\pi}_{N/2}}$ and $y'_{N/2}, \boxed{\boldsymbol{\pi}'_{N/2}}$

Proof size: $P(N) = \boldsymbol{P(N/2)}$

 # Steps: $T(N) = 2\,T\,(N/2) + \mathbf{1}$

Merge Proofs [Valiant'06]



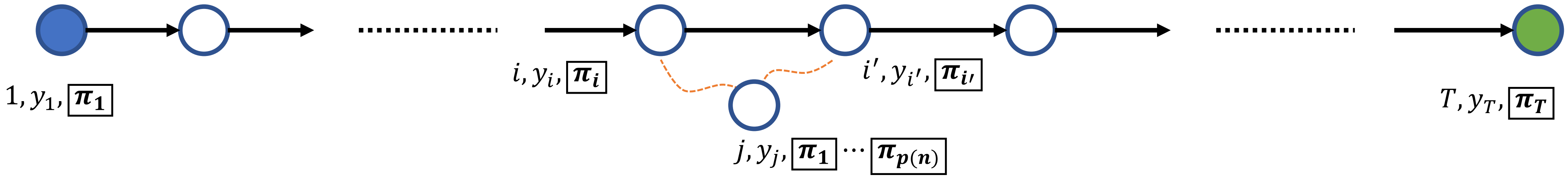
Merge proofs for $y_{N/2}, \boxed{\boldsymbol{\pi}_{N/2}}$ and $y'_{N/2}, \boxed{\boldsymbol{\pi}'_{N/2}}$

Proof size: $P(N) = \boldsymbol{P(N/2)}$

 # Steps: $\boldsymbol{T}(\boldsymbol{N}) = \boldsymbol{2\,T\,(N/2) + 1}$

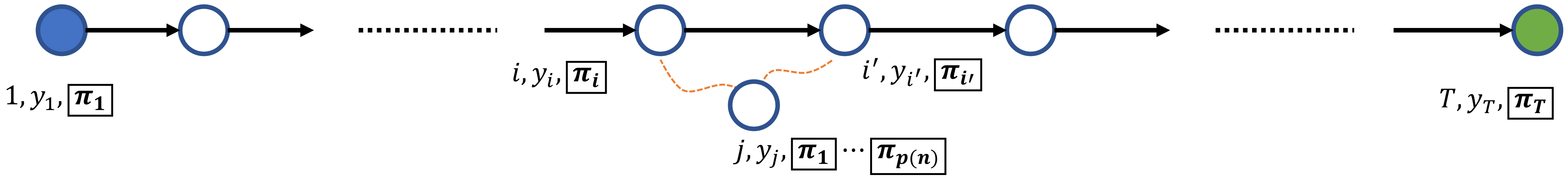
Requires "super-extractable" SNARKs

New Idea: Incremental Merge



Merge via (long) incrementally verifiable computation.

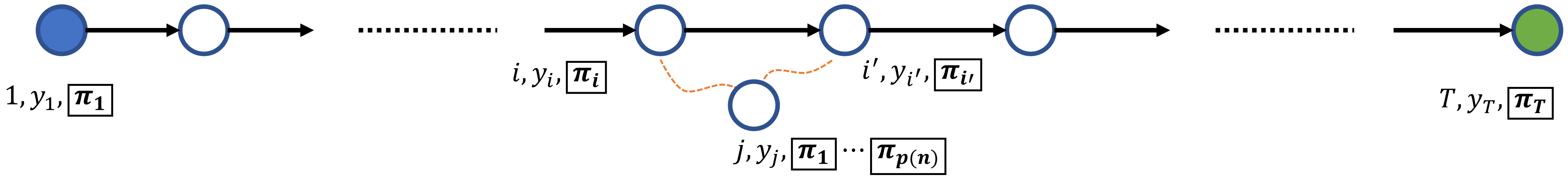
New Idea: Incremental Merge



Merge via (long) incrementally verifiable computation.

How long?

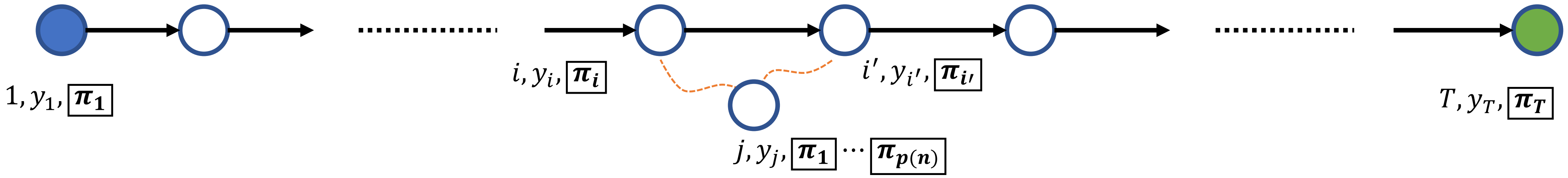
New Idea: Incremental Merge



Merge via (long) incrementally verifiable computation.

How long? $\boldsymbol{O(T(N/2))}$

New Idea: Incremental Merge



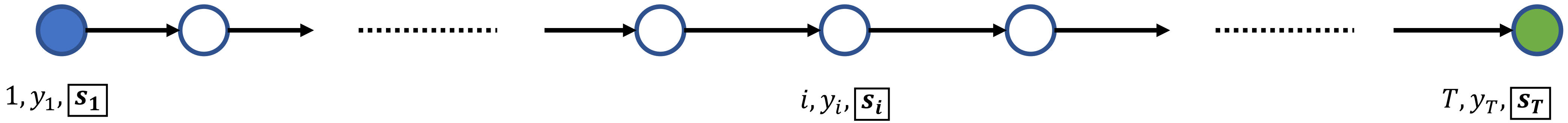
Merge via (long) incrementally verifiable computation.

How long? $\boldsymbol{O(T(N/2))}$

Proof size: $\boldsymbol{P(N) = P(N/2) + poly(n)}$

 # Steps: $\boldsymbol{T(N) = d\,T\,(N/2) + poly(n)}$

New Idea: Incremental Merge



How do you efficiently compute

$$\boldsymbol{s}\left(\boldsymbol{i},\ \boldsymbol{y_i}, \boxed{\boldsymbol{\pi_i}}\right) = \boldsymbol{i} + \mathbf{1},\ \boldsymbol{y_{i+1}}, \boxed{\boldsymbol{\pi_{i+1}}}$$

Proof Merging for (Fiat-Shamir) sumcheck

$$\sum_{z_{j+1},\cdots,z_n\in\{0,1\}} f(\beta_1,\cdots,\beta_{j-1},\beta_j,z_{j+1},\cdots,z_n) = y_j$$

Proof Merging for (Fiat-Shamir) sumcheck

$$\sum_{z_{j+1},\cdots,z_n\in\{0,1\}} f(\beta_1,\cdots,\beta_{j-1},\beta_j,z_{j+1},\cdots,z_n) = y_j$$

$$\xrightarrow{\tilde{g}_{j+1}(x)}$$

Proof Merging for (Fiat-Shamir) sumcheck

$$\sum_{z_{j+1},\cdots,z_n \in \{0,1\}} f(\beta_1, \cdots, \beta_{j-1}, \beta_j, z_{j+1}, \cdots, z_n) = y_j$$

$$\tilde{g}_{j+1}(x)$$

$$\beta_{j+1} = h(\alpha_1, \beta_1, \cdots, \alpha_{j+1})$$

Proof Merging for (Fiat-Shamir) sumcheck

$$\sum_{z_{j+1},\cdots,z_n \in \{0,1\}} f(\beta_1,\cdots,\beta_{j-1},\beta_j,z_{j+1},\cdots,z_n) = y_j$$

$\tilde{g}_{j+1}(x)$

$y_{j+1} \coloneqq \tilde{g}_{j+1}(\beta_{j+1})$

$\beta_{j+1} = h(\alpha_1,\beta_1,\cdots,\alpha_{j+1})$

$j+1$-th claim

Proof Merging for (Fiat-Shamir) sumcheck

$$\sum_{z_{j+1},\cdots,z_n \in \{0,1\}} f(\beta_1,\cdots,\beta_{j-1},\beta_j,z_{j+1},\cdots,z_n) = y_j$$

$\tilde{g}_{j+1}(x)$

$\beta_{j+1} = h(\alpha_1,\beta_1,\cdots,\alpha_{j+1})$

$y_{j+1} \coloneqq \tilde{g}_{j+1}(\beta_{j+1})$

$j+1$-th claim

size $N/2^{j+1}$ claim

Proof Merging for (Fiat-Shamir) sumcheck

$$\sum_{z_{j+1},\cdots,z_n \in \{0,1\}} f(\beta_1, \cdots, \beta_{j-1}, \beta_j, z_{j+1}, \cdots, z_n) = y_j$$

$$\{\tilde{g}_{j+1}(0), \cdots, \tilde{g}_{j+1}(d)\}$$

$$y_{j+1} \coloneqq \tilde{g}_{j+1}(\beta_{j+1})$$

$$\beta_{j+1} = h(\alpha_1, \beta_1, \cdots, \alpha_{j+1})$$

$j+1$-th claim

size $N/2^{j+1}$ claim

Proof Merging for (Fiat-Shamir) sumcheck

$$\sum_{z_{j+1},\cdots,z_n\in\{0,1\}} f(\beta_1,\cdots,\beta_{j-1},\beta_j,z_{j+1},\cdots,z_n) = y_j$$

size $N/2^{j+1}$ claims

$\{\tilde{g}_{j+1}(0),\cdots,\tilde{g}_{j+1}(d)\}$

$\beta_{j+1} = h(\alpha_1,\beta_1,\cdots,\alpha_{j+1})$

$y_{j+1} \coloneqq \tilde{g}_{j+1}(\beta_{j+1})$

$j+1$-th claim

size $N/2^{j+1}$ claim

Proof Merging for (Fiat-Shamir) sumcheck

$$\sum_{z_{j+1},\cdots,z_n\in\{0,1\}} f(\beta_1,\cdots,\beta_{j-1},\beta_j,z_{j+1},\cdots,z_n) = y_j$$

size $N/2^{j+1}$ claims

$\{\tilde{g}_{j+1}(0),\cdots,\tilde{g}_{j+1}(d)\}$

$y_{j+1} \coloneqq \tilde{g}_{j+1}(\beta_{j+1})$

$\beta_{j+1} = h(\alpha_1,\beta_1,\cdots,\alpha_{j+1})$

$j+1$-th claim

size $N/2^{j+1}$ claim

$(d+2)$ size $N/2^{j+1}$ claims

Proving the j-th claim

$$\left\{ \begin{array}{c} \tilde{g}_{j+1}(0), \boxed{\boldsymbol{\pi_0}} \\ \tilde{g}_{j+1}(1), \boxed{\boldsymbol{\pi_1}} \\ \vdots \\ \tilde{g}_{j+1}(d), \boxed{\boldsymbol{\pi_d}} \end{array} \right.$$

Proving the j-th claim

$$\left\{\begin{array}{l} \tilde{g}_{j+1}(0), \boxed{\boldsymbol{\pi_0}} \\ \tilde{g}_{j+1}(1), \boxed{\boldsymbol{\pi_1}} \\ \vdots \\ \tilde{g}_{j+1}(d), \boxed{\boldsymbol{\pi_d}} \end{array}\right.$$

size $N/2^{j+1}$ claims

Proving the j-th claim

$$\xrightarrow{\tilde{\boldsymbol{g}}_{\boldsymbol{j+1}}(\boldsymbol{x})} \left\{ \begin{array}{l} \tilde{g}_{j+1}(0), \boxed{\boldsymbol{\pi_0}} \\ \tilde{g}_{j+1}(1), \boxed{\boldsymbol{\pi_1}} \\ \vdots \\ \tilde{g}_{j+1}(d), \boxed{\boldsymbol{\pi_d}} \end{array} \right.$$

size $\boldsymbol{N/2^{j+1}}$ claims

Proving the j-th claim

$\tilde{\boldsymbol{g}}_{j+1}(\boldsymbol{x})$

$\tilde{g}_{j+1}(0), \boxed{\boldsymbol{\pi_0}}$

$\tilde{g}_{j+1}(1), \boxed{\boldsymbol{\pi_1}}$

$\vdots$

$\tilde{g}_{j+1}(d), \boxed{\boldsymbol{\pi_d}}$

size $N/2^{j+1}$ claims

$\boldsymbol{\beta_{j+1} = h(\alpha_1, \beta_1, \cdots, \alpha_{j+1})}$

Proving the j-th claim

$$\xrightarrow{\boldsymbol{\tilde{g}_{j+1}(x)}}$$

$$\left\{ \begin{array}{l} \tilde{g}_{j+1}(0), \boxed{\boldsymbol{\pi_0}} \\ \tilde{g}_{j+1}(1), \boxed{\boldsymbol{\pi_1}} \\ \vdots \\ \tilde{g}_{j+1}(d), \boxed{\boldsymbol{\pi_d}} \end{array} \right.$$

size $N/2^{j+1}$ claims

$$\xleftarrow{\boldsymbol{\beta_{j+1} = h(\alpha_1, \beta_1, \cdots, \alpha_{j+1})}}$$

$$\tilde{g}_{j+1}(\beta_{j+1}), \boxed{\boldsymbol{\pi_{d+1}}}$$

Proving the j-th claim

$\tilde{\boldsymbol{g}}_{\boldsymbol{j+1}}(\boldsymbol{x})$

$\tilde{g}_{j+1}(0), \boxed{\boldsymbol{\pi_0}}$

$\tilde{g}_{j+1}(1), \boxed{\boldsymbol{\pi_1}}$

$\vdots$

$\tilde{g}_{j+1}(d), \boxed{\boldsymbol{\pi_d}}$

size $N/2^{j+1}$ claims

$\boldsymbol{\beta_{j+1} = h(\alpha_1, \beta_1, \cdots, \alpha_{j+1})}$

$\tilde{g}_{j+1}(\beta_{j+1}), \boxed{\boldsymbol{\pi_{d+1}}}$

size $N/2^{j+1}$ claim

Parameters

Parameters

Proof size: $P(N) = \boldsymbol{P(N/2) + log}|\mathbb{F}|$

Parameters

Proof size: $\boldsymbol{P(N) = P(N/2) + log|\mathbb{F}|}$

$$\boldsymbol{P(0) = log|\mathbb{F}|}$$

Parameters

Proof size: $\boldsymbol{P(N) = P(N/2) + log|\mathbb{F}|}$

$$\boldsymbol{P(0) = log|\mathbb{F}|}$$

$$\boldsymbol{P(n) = poly(n)}$$

Parameters

Proof size: $P(N) = P(N/2) + log|\mathbb{F}|$

Steps: $T(N) = (d+2)\, T\,(N/2) + poly(n)$

$$P(0) = log|\mathbb{F}|$$

$$P(n) = poly(n)$$

Parameters

Proof size: $\boldsymbol{P(N) = P(N/2) + log|\mathbb{F}|}$

 # Steps: $\boldsymbol{T(N) = (d+2)\, T\,(N/2) + poly(n)}$

$$\boldsymbol{P(0) = log|\mathbb{F}|}$$

$$\boldsymbol{P(n) = poly(n)}$$

$$\boldsymbol{T(0) = poly(log|\mathbb{F}|)}$$

Parameters

Proof size: $P(N) = P(N/2) + log|\mathbb{F}|$

 # Steps: $T(N) = (d+2)\,T\,(N/2) + poly(n)$

$P(0) = log|\mathbb{F}|$

$P(n) = poly(n)$

$T(0) = poly(log|\mathbb{F}|)$

$T(n) = 2^{O(n)}$

Open Problems

Instantiating Fiat-Shamir for sumcheck

Sampling small(ish) hard instances of NASH

Thank you. Questions?

The Five Worlds of Impagliazzo

Cryptomania:
∃ public-key crypto

Minicrypt:
∃ one-way functions ∄
public-key crypto

Pessiland:
NP is hard-on-average ∄
one-way functions

Heuristica:
$P \neq NP$
NP is easy-on-average

Algorithmica:
$P = NP$

The Five Worlds of Impagliazzo

Cryptomania:
∃ public-key crypto

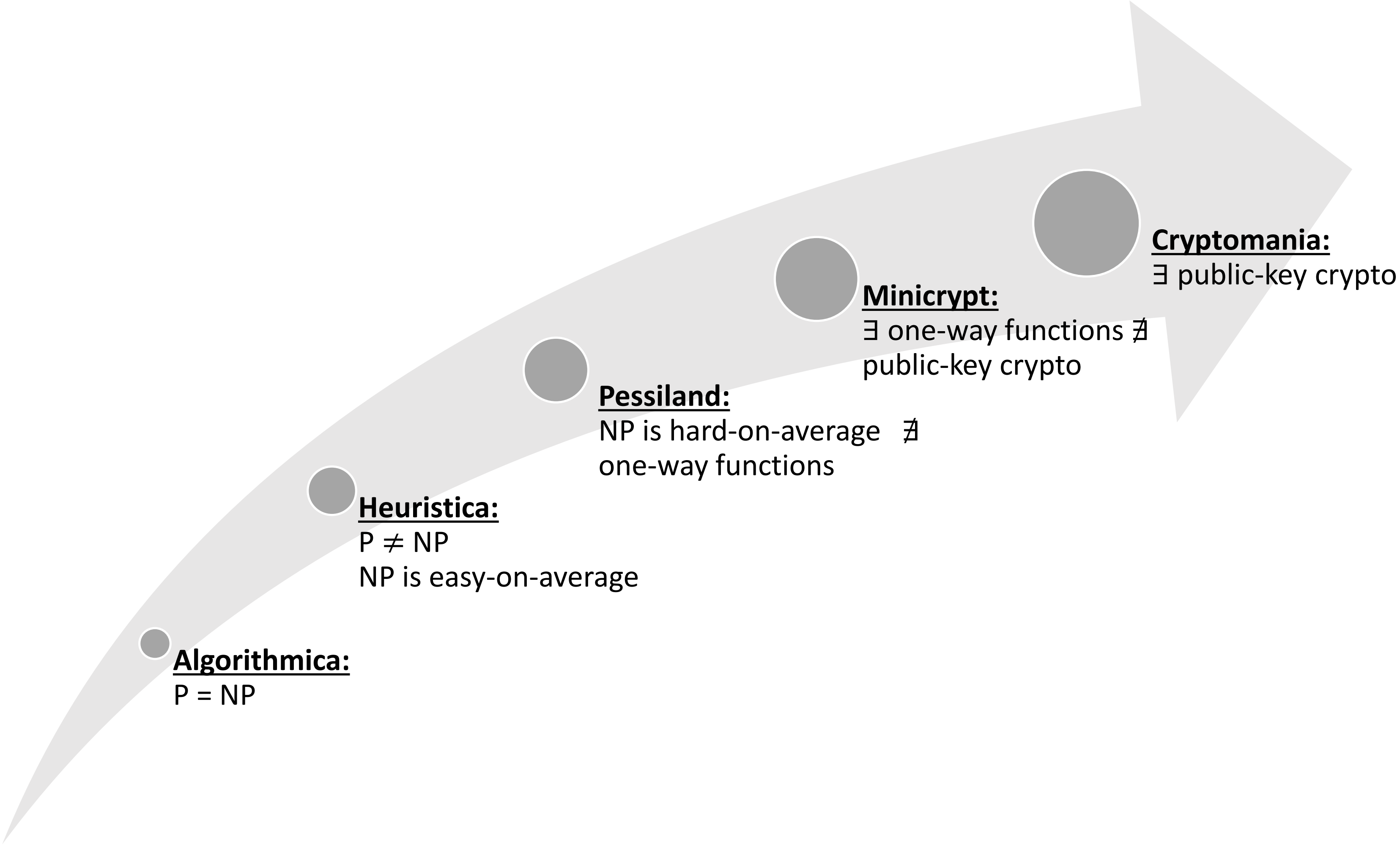
Minicrypt:
∃ one-way functions ∄
public-key crypto

Pessiland:
NP is hard-on-average ∄
one-way functions

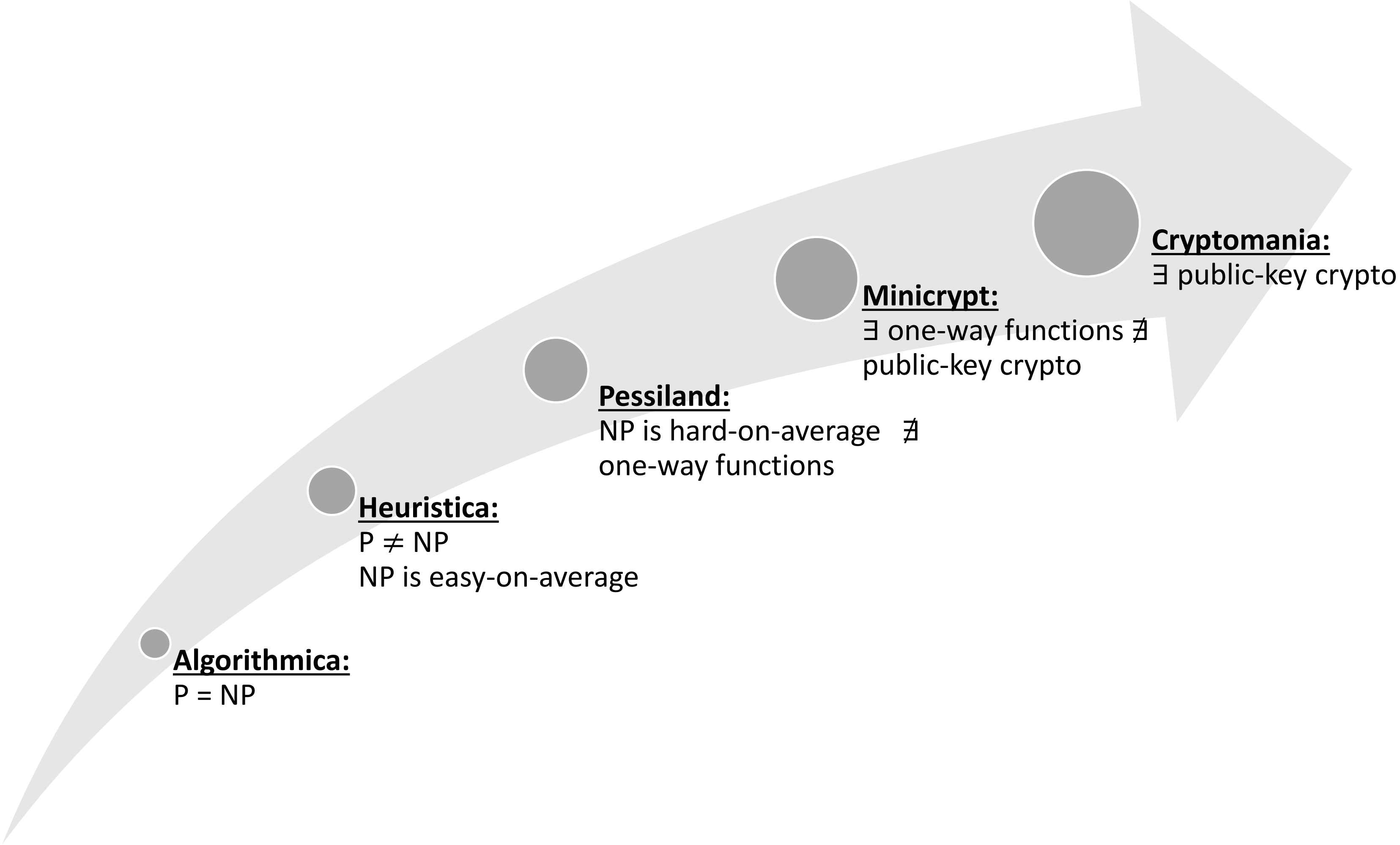
Heuristica:
P ≠ NP
NP is easy-on-average

Algorithmica:
P = NP

The Five Worlds of Impagliazzo



The Five Worlds of Impagliazzo



The Five Worlds of Impagliazzo

Algorithmica:
P = NP

Heuristica:
P ≠ NP
NP is easy-on-average

Pessiland:
NP is hard-on-average ∄
one-way functions

Minicrypt:
∃ one-way functions ∄
public-key crypto

Cryptomania:
∃ public-key crypto

The Five Worlds of Impagliazzo

Algorithmica:
P = NP

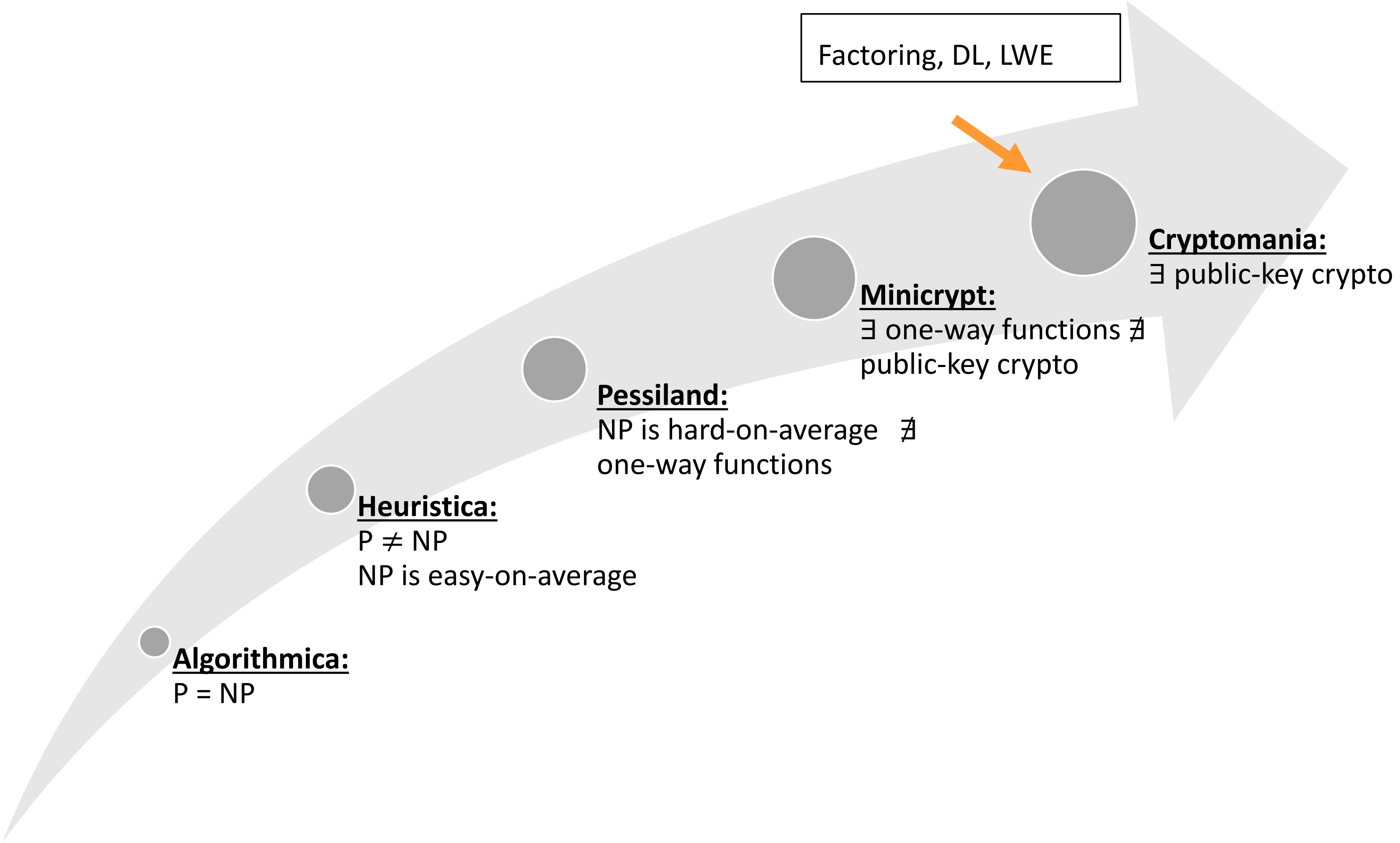
Heuristica:
P ≠ NP
NP is easy-on-average

Pessiland:
NP is hard-on-average ∄
one-way functions

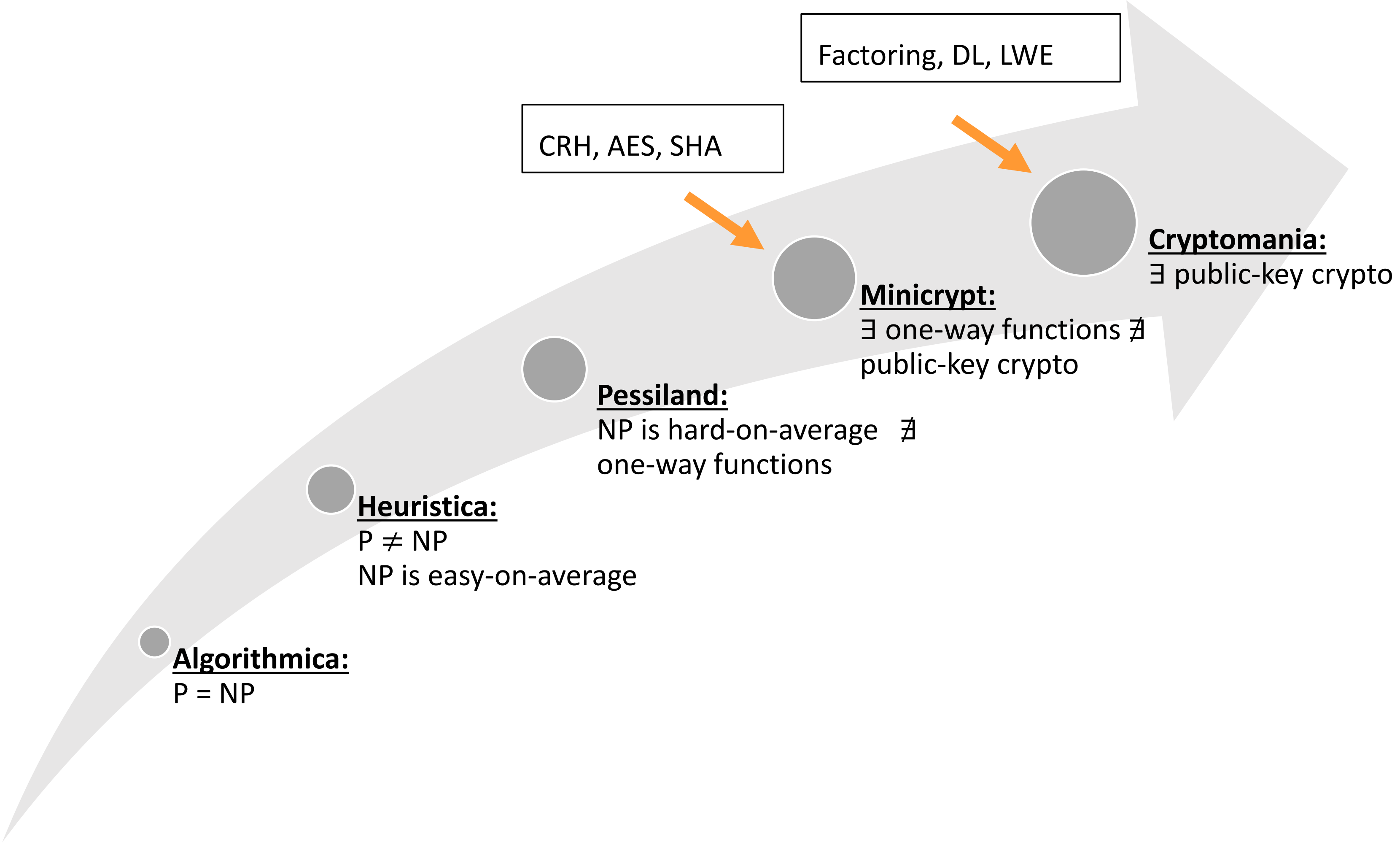
Minicrypt:
∃ one-way functions ∄
public-key crypto

Cryptomania:
∃ public-key crypto

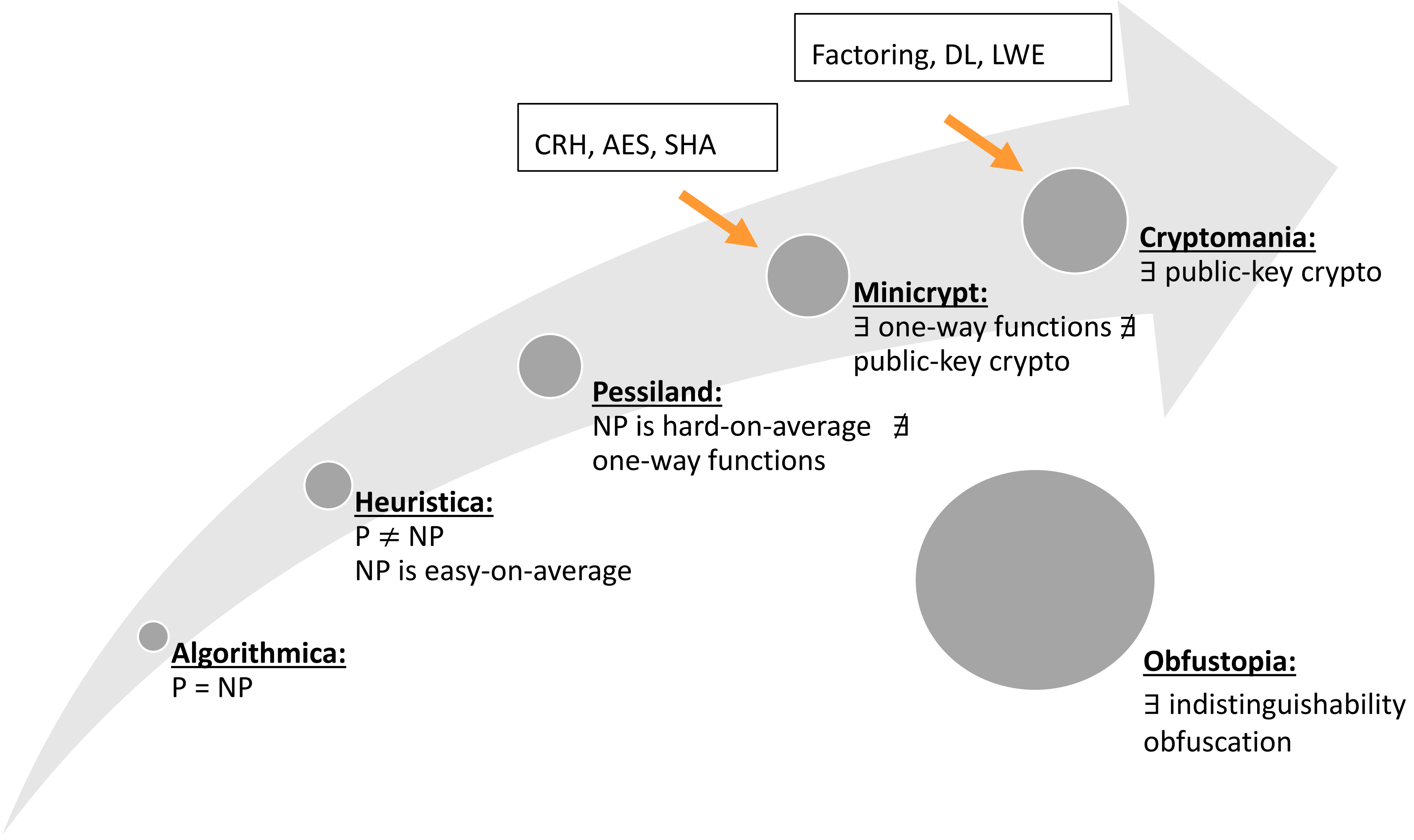
The Five Worlds of Impagliazzo



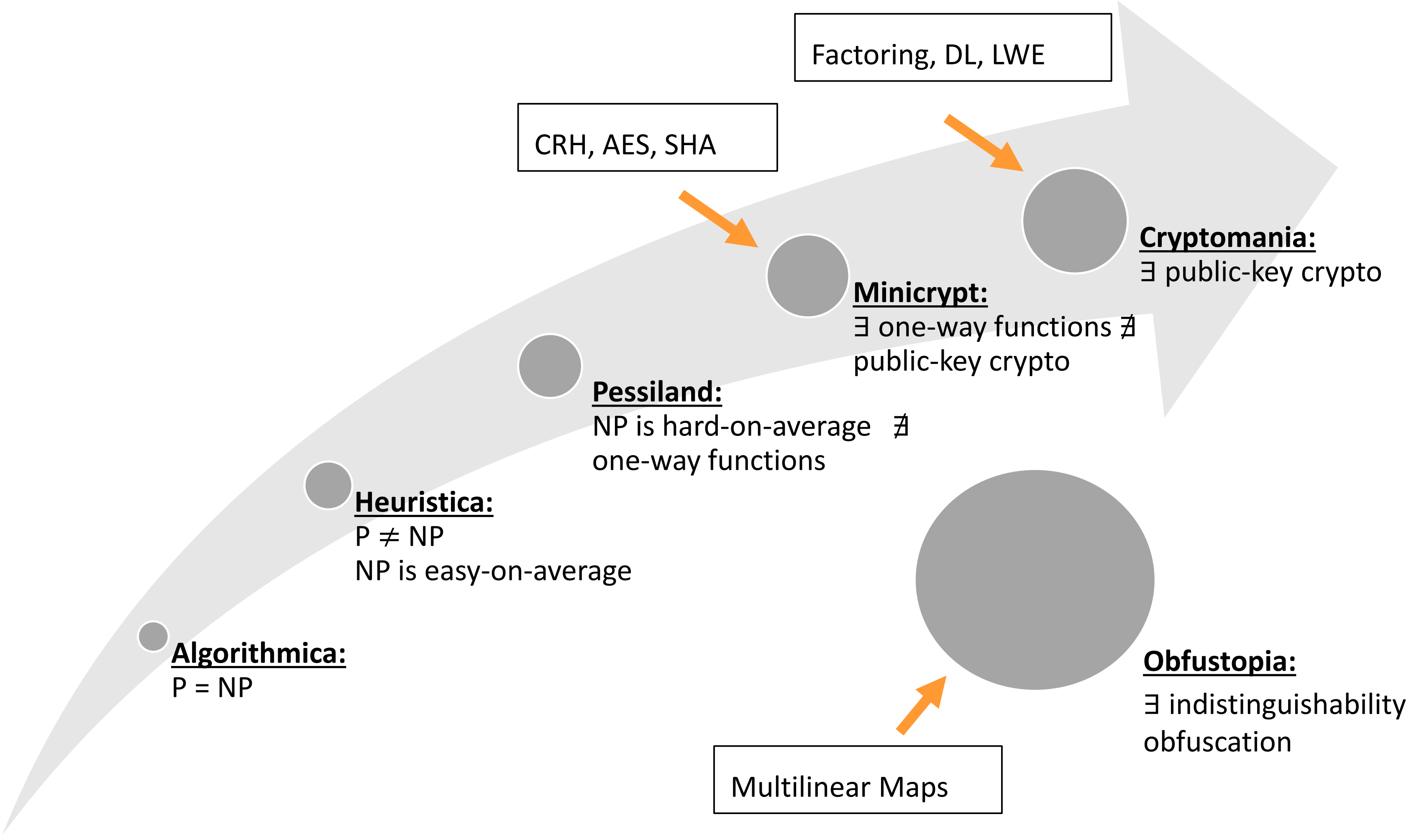
The Five Worlds of Impagliazzo



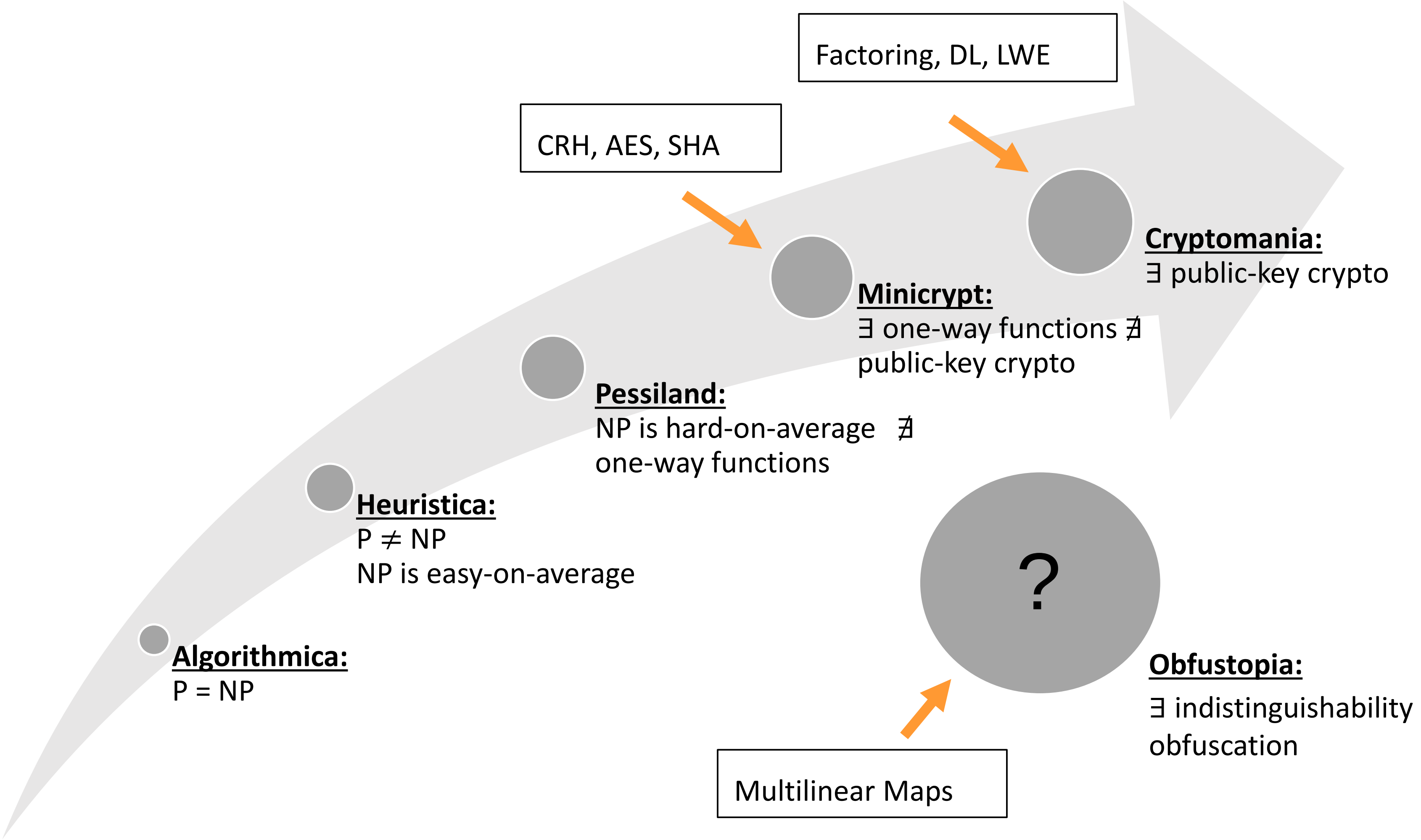
The Five Worlds of Impagliazzo



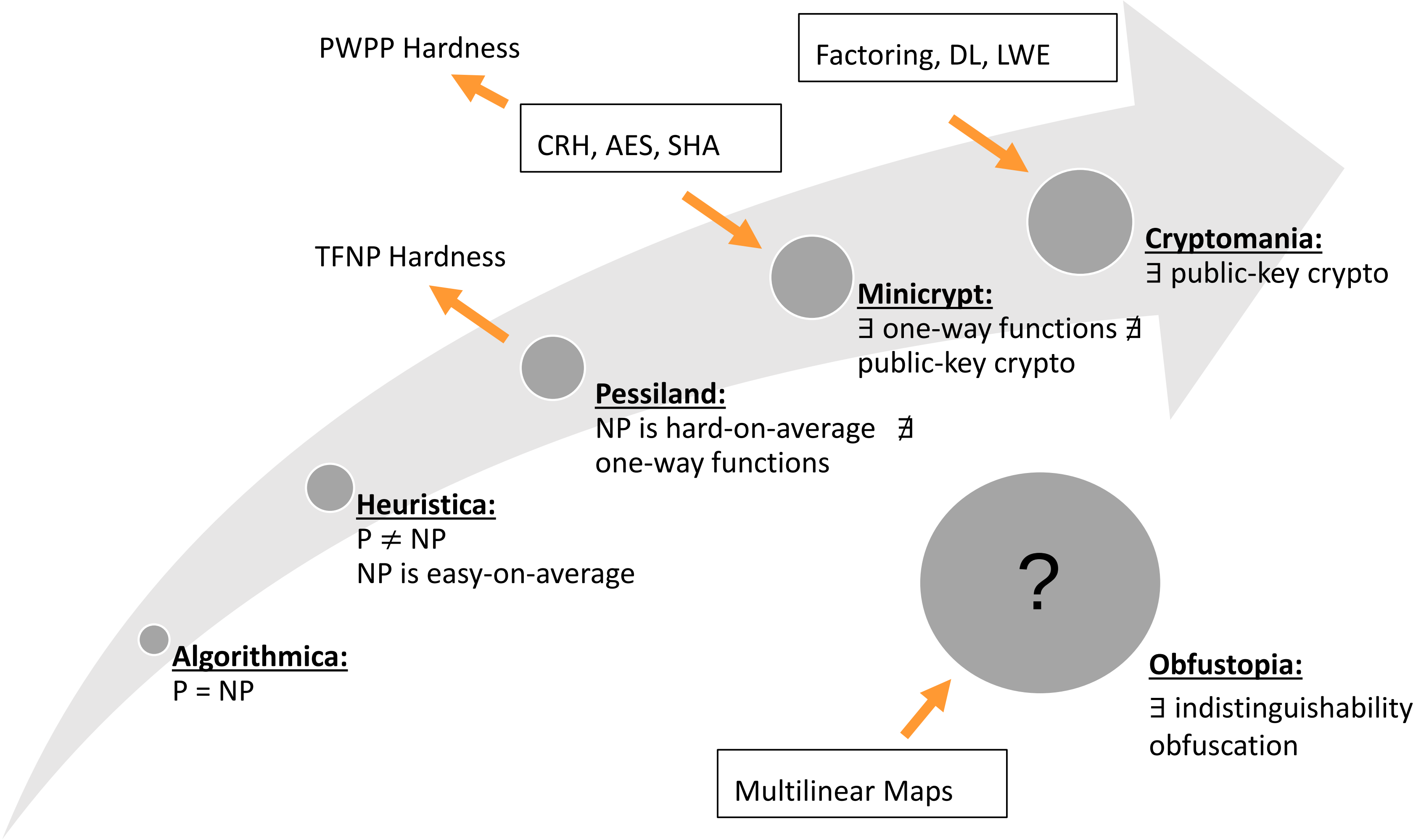
The Five Worlds of Impagliazzo



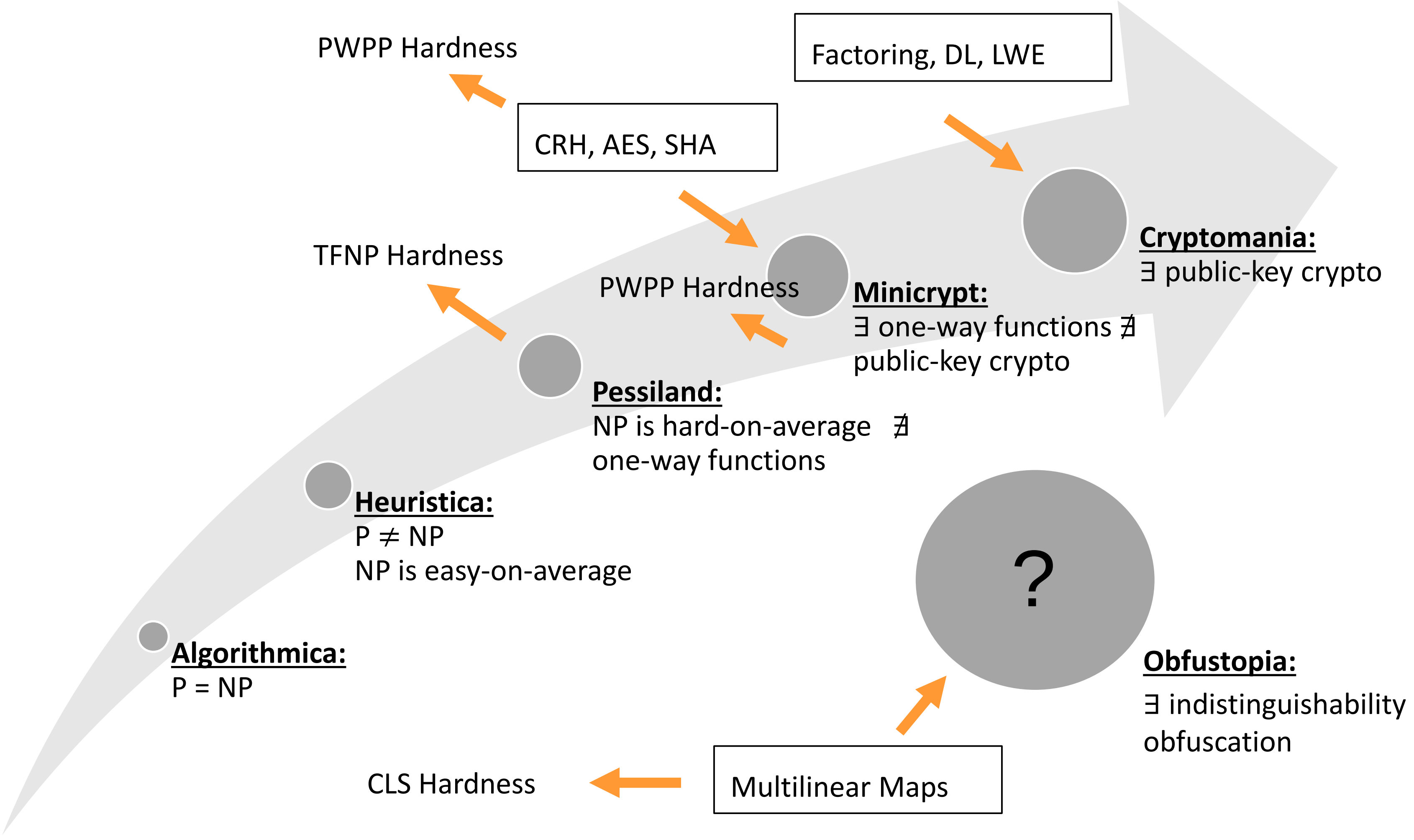
The Five Worlds of Impagliazzo



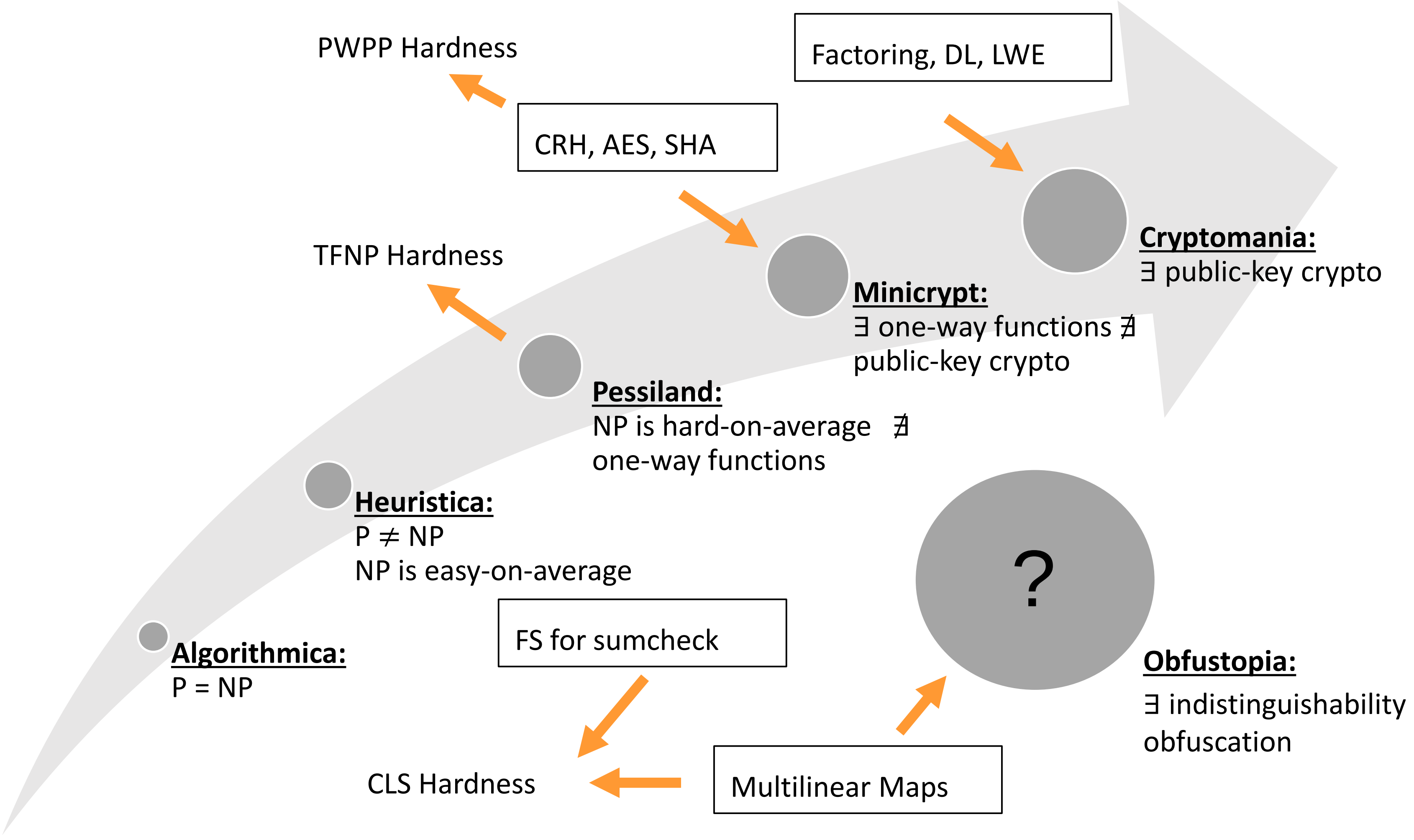
The Five Worlds of Impagliazzo



The Five Worlds of Impagliazzo



The Five Worlds of Impagliazzo



Is Crypto hardness Necessary?

[Rosen-Segev-Shachaf'17]

Black box separations

SVL hardness **not essential** for PPAD hardness

Basing PPAD hardness on OWFs:

cannot go through SVL, and

must have **exponential** #sol