Efficient Recursion via Statement Folding

zk-School
20 Apr 2023

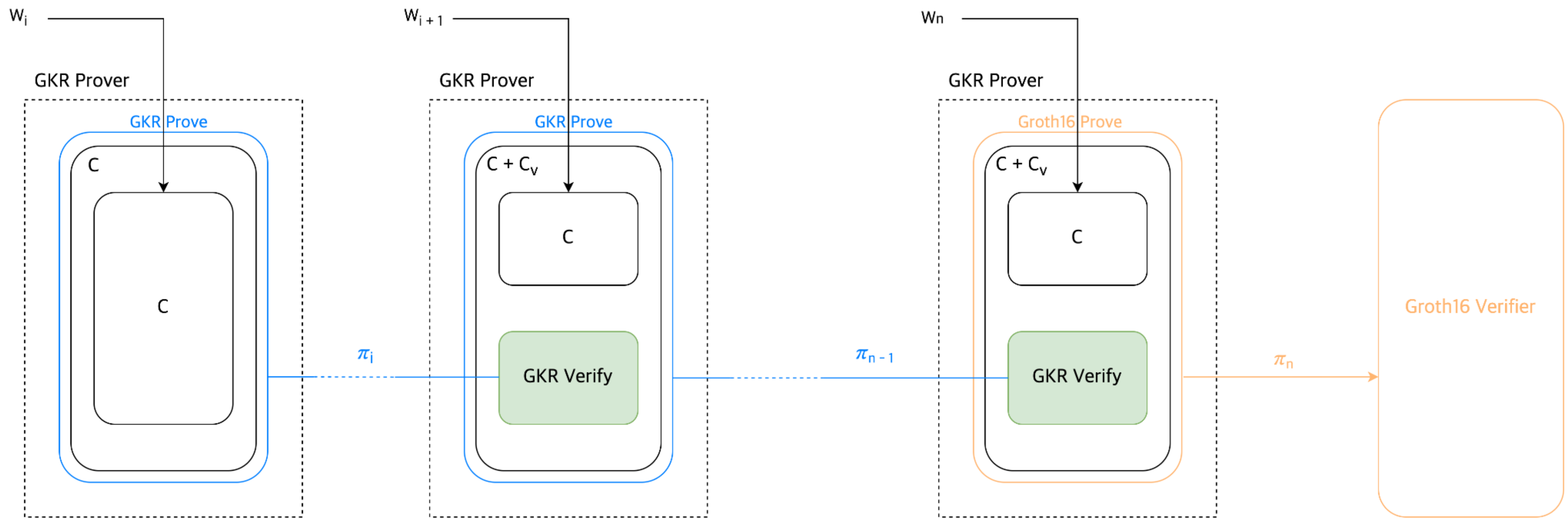
contents

- review for recursive proof
- IVC
- Nova

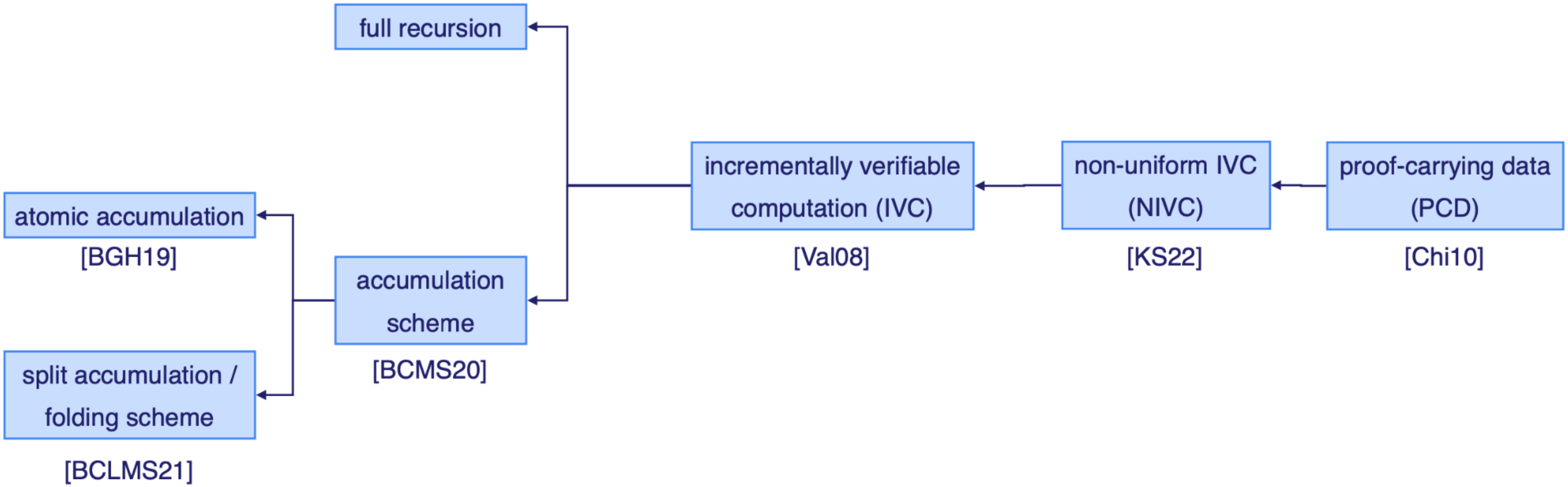
Recursive proof

- Recursion: 재귀
- 재귀: 정의(definition) 안에서 자기 자신이 참조되는 것
- 재귀 증명: 증명 과정 안에서 자기 자신의 검증 과정을 포함시킴
- 증명을 생성 후 다음 증명을 생성할 때 이전 단계의 증명을 검증
- 한번의 검증으로 이전 단계의 모든 증명이 검증

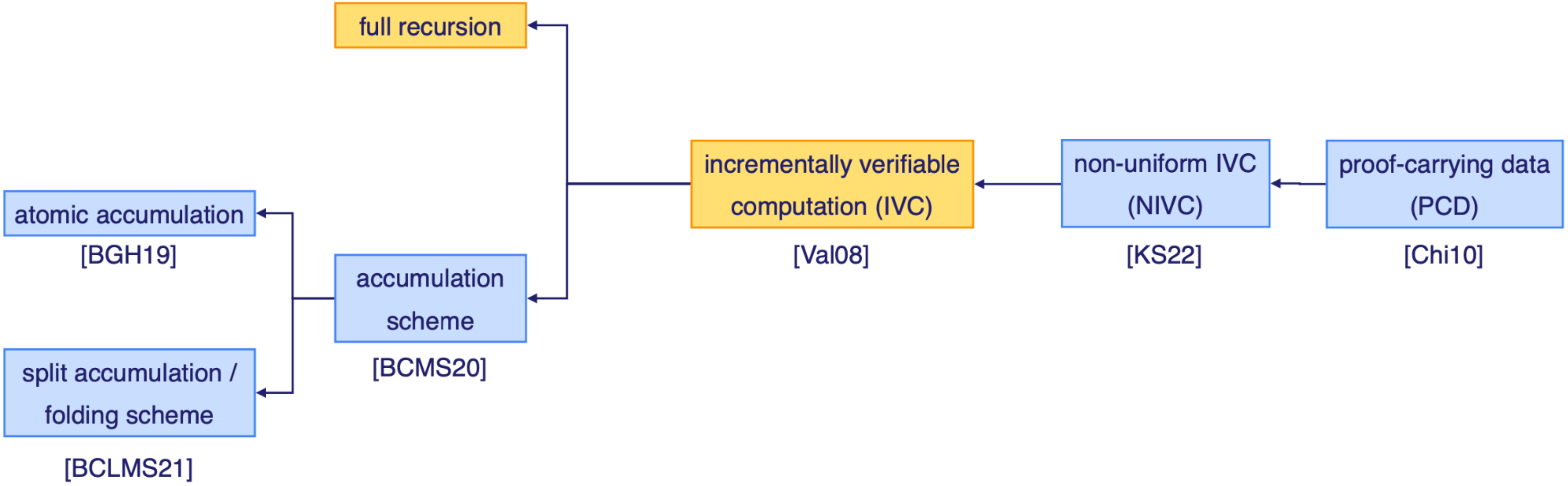
Recursive proof



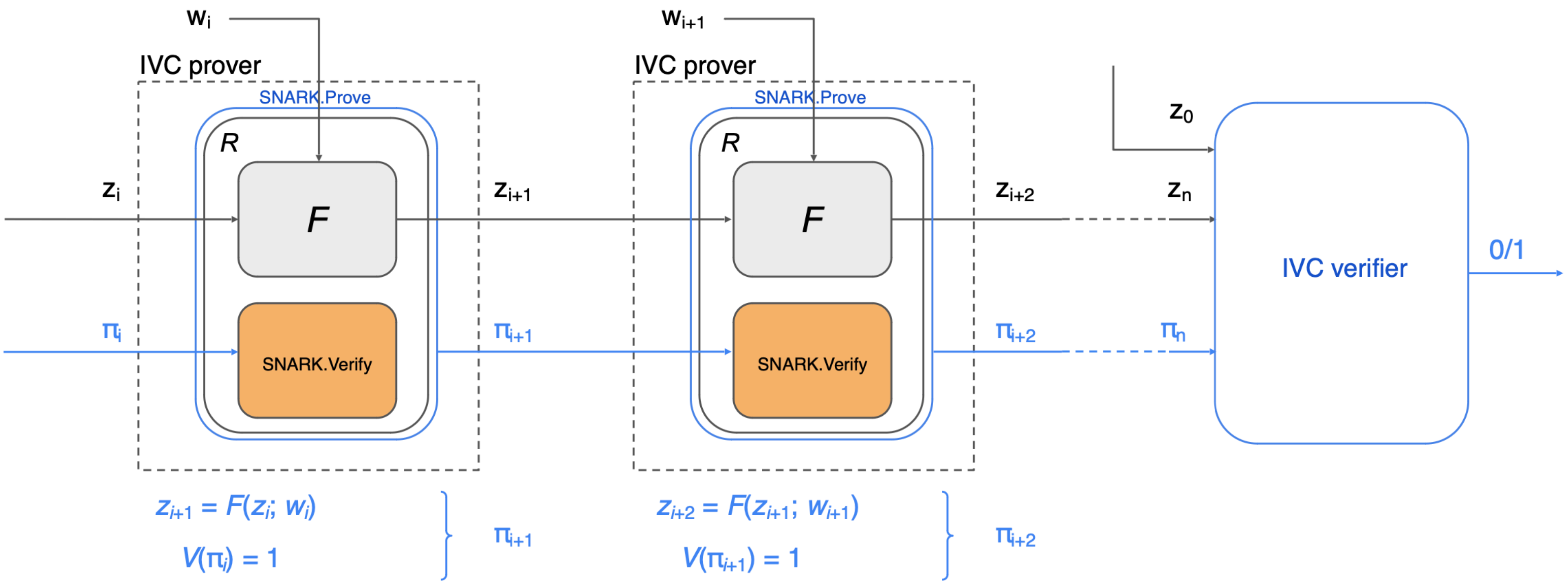
Recursive proof



Recursive proof



IVC



w_i
IVC prover
SNARK.Prove
R
F
SNARK.Verify
z_i
π_i
z_{i+1}
π_{i+1}
w_{i+1}
IVC prover
SNARK.Prove
R
F
SNARK.Verify
z_{i+2}
π_{i+2}
z_0
z_n
π_n
IVC verifier
0/1
$z_{i+1} = F(z_i; w_i)$
$V(\pi_i) = 1$
π_{i+1}
$z_{i+2} = F(z_{i+1}; w_{i+1})$
$V(\pi_{i+1}) = 1$
π_{i+2}

IVC[Valiant'08]

- 증명자가 $\omega_1, \ldots, \omega_n$으로부터 계산되는 s_n을 정확히 계산했다는 것을 증명
- 하나의 큰 연산이나 작동을 여러 단계로 나눈 뒤 각각의 단계에서의 계산이 정확하다는 것을 증명
- $i = 1,...,n$ 에서 i번째 증명자는 계산 결과 s_i와 증명 π_i를 결과값으로 냄
- π_i := 증명자가 다음을 만족하는 $(s_{i-1}, \omega_i, \pi_{i-1})$를 가지고 있음
  - $F(s_{i-1}, \omega_i) = s_i$
  - $V(vp, (i-1, s_0, s_{i-1}), \pi_{i-1}) =$ yes

IVC의 활용

- F := VM(EVM, Risc5 등)에서 하나의 step
- 증명자는 매우 큰 프로그램을 한번에 증명할 때보다 훨씬 작은 메모리 사용으로 프로그램이 정확하게 실행되었다는 것을 증명 가능

IVC의 활용

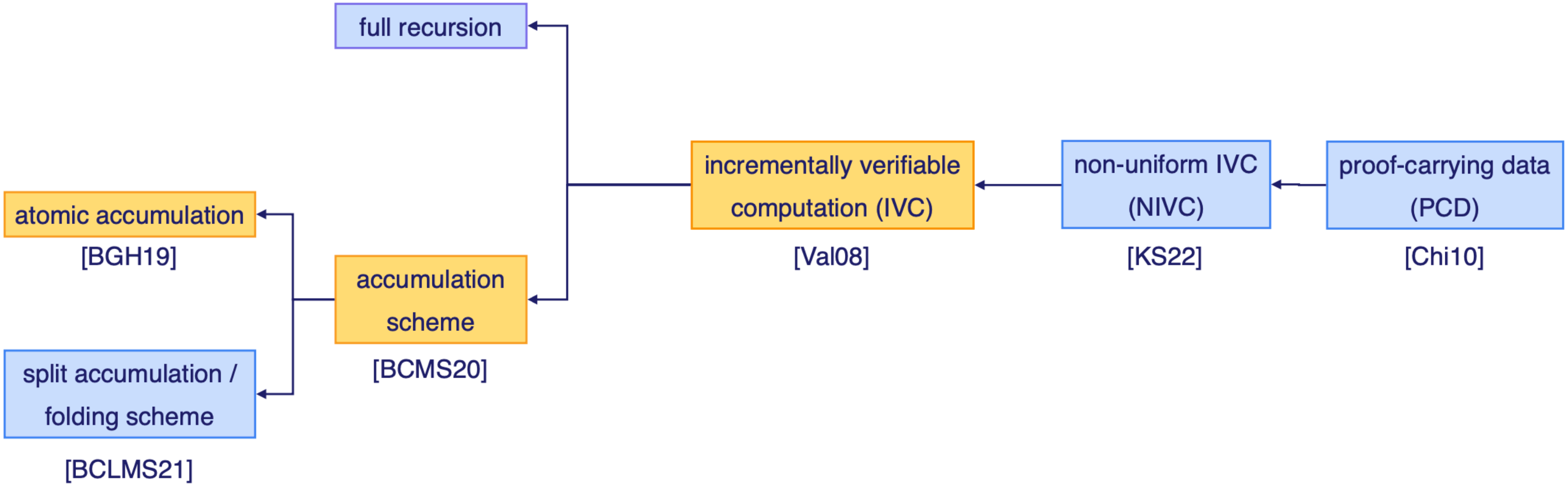
- F := 블록체인 tx을 처리하는 로직
- 블록체인의 state가 여러 tx에 따라 바뀌지만 마지막 증명으로 검증 가능

Full recursion

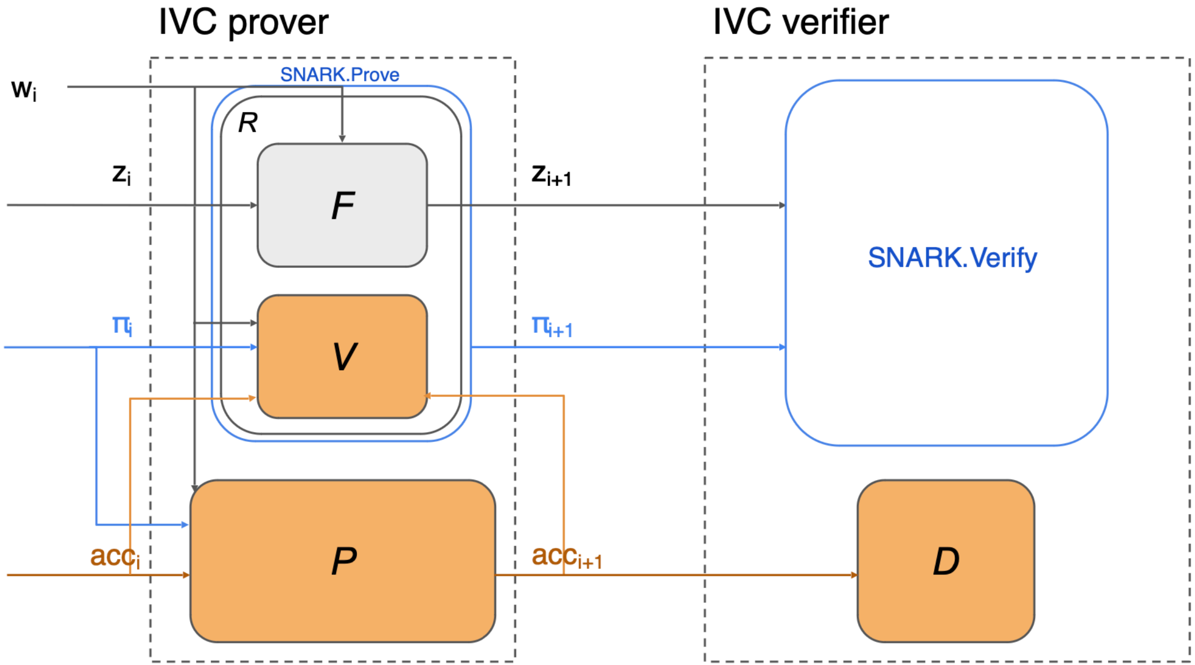
- 매우 비쌈 (시간, 공간)
- 증명자가 검증 연산을 하는 것을 증명으로 만드는 것

☑ 검증 연산의 대부분을 circuit 외부에서 진행

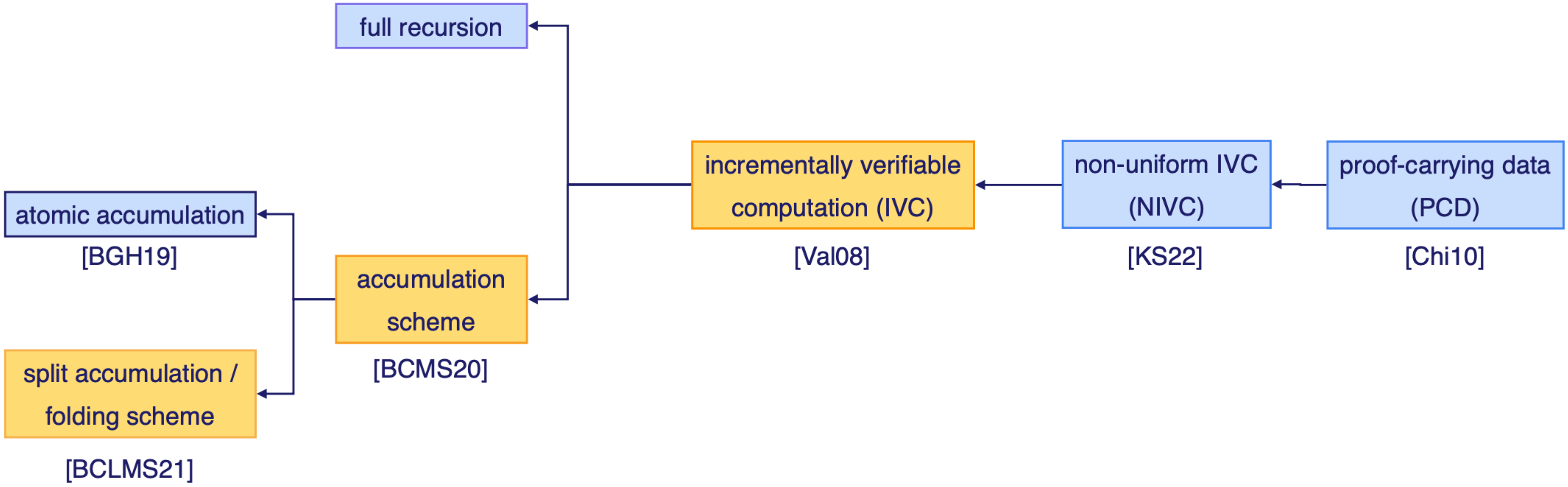
atomic accumulation



atomic accumulation



Folding scheme



Folding scheme

- idea: 여러 증명이 있다는 것은 각각 여러 instance에서 증명을 생성했다는 것이다.
  그렇다면 특정 연산을 통해 증명 생성 전에 여러 instance 들을 합치면 한번의 증명 생성으로 여러 instance를 증명할 수 있지 않을까?

Folding scheme

- R1CS $A, B, C \in \mathbb{F}_p^{u \times v}$
    - instance 1: public x_1, witness $z_1 = (x_1, w_1) \in \mathbb{F}_p^v$
    - instance 2: public x_2, witness $z_2 = (x_2, w_2) \in \mathbb{F}_p^v$
- $(Az_i) \circ (Bz_i) = Cz_i$ for $i = 1,2$

Folding scheme

Attempt 1

- $x \leftarrow x_1 + rx_2$ (r은 검증자가 무작위로 선택한 값)
- $z \leftarrow z_1 + rz_2 = (x_1 + rx_2, w_1 + rw_2)$
- R1CS 형태로 표현 불가능
- $(Az) \circ (Bz) = A(z_1 + rz_2) \circ B(z_1 + rz_2)$

$E \in \mathbb{F}_p^u$

$$= A(z_1) \circ B(z_1) + r^2(Az_2) \circ (Bz_2) \boxed{+ r(Az_2) \circ (Bz_1) + r(Az_1) \circ (Bz_2)}$$

$$= Cz_1 + r^2 Cz_2 + E$$

Relaxed R1CS

- $(Az) \circ (Bz) = Cz \rightarrow (Az) \circ (Bz) = u(Cz) + E$
- $u = 1,\ E = 0$일 때 original R1CS
- cross term을 $T = (Az_2) \circ (Bz_1) + (Az_1) \circ (Bz_2) - u_1(Cz_2) - u_2(Cz_1)$로 전달
- $x \leftarrow x_1 + rx_2$, $u \leftarrow u_1 + ru_2$, $E \leftarrow E_1 + rT + r^2E_2$
- $(Az) \circ (Bz) =$

$$= (Az_1) \circ (Bz_1) + \mathrm{r}^2\,(Az_2) \circ (Bz_2) + \boxed{r(Az_2) \circ (Bz_1) + r(Az_1) \circ (Bz_2)}$$

$$= \overbrace{c_1(Dz_1) + E_1} + \overbrace{r^2c_2(Dz_2) + r^2E_2} + r\boxed{[(Az_2) \circ (Bz_1) + (Az_1) \circ (Bz_2)]}$$

$$= (c_1 + rc_2)(Dz_1 + rDz_2) + \underbrace{\mathrm{E}_1 + r^2\mathrm{E}_2 + rT}_{E}$$

$$= c(Dz) + E$$

Committed Relaxed R1CS

- Relaxed R1CS는 zk가 아님 + E가 클 수 있음
- 검증자는 $(x, u, commit(E, r_E))$, 증명자는 (z, E, r_E)를 가지고 있음
- homomorphic commitments
- $commit(m_1, r_1) + commit(m_2, r_2) = commit(m_1 + m_2, r_1 + r_2)$

Committed Relaxed R1CS

- $T \leftarrow (Az_2) \circ (Bz_1) + (Az_1) \circ (Bz_2) - u_1(Cz_2) - u_2(Cz_1)$
- 무작위 값 r_T를 통해 $com_T \leftarrow commit(T, r_T)$를 검증자에게 보냄
- 검증자는 무작위 값 r을 선택 후 증명자에게 보냄
- $x \leftarrow x_1 + rx_2,\ u \leftarrow u_1 + ru_2,\ com_E \leftarrow com_{E_1} + r \cdot com_T + r^2 \cdot com_{E_2}$

Nova

- Folding scheme
- + Fiat-Shamir (to non-interactive)