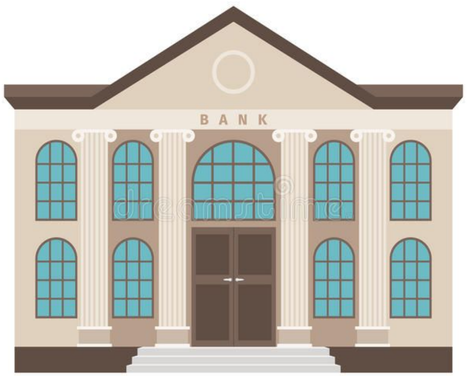
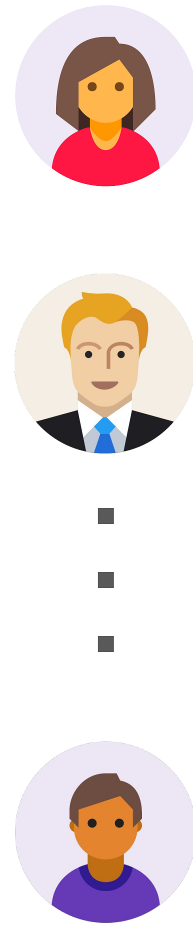
Scaling Transaction Verifications in Cryptocurrencies

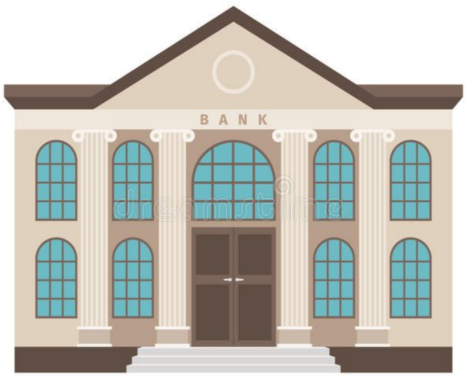
PRIMES Computer Science Conference
October 13th, 2018
Yiming Zheng
Alin Tomescu

Motivation





Motivation



Balance: $50

Balance: $40

Balance: $50

Motivation

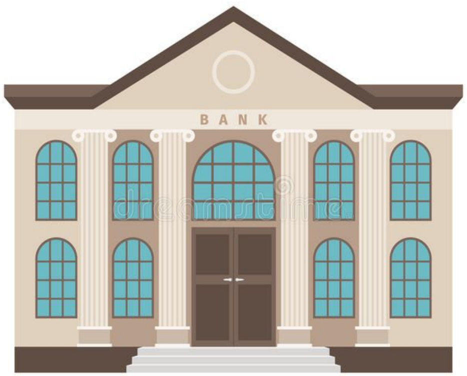
Digest: d_n

Balance: $50

Balance: $40

Balance: $50

Motivation



Digest: d_n

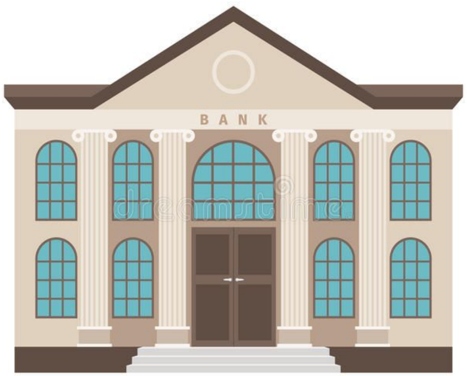
Balance: \$50,
Proof: $\Pi_A(n)$

Balance: \$40,
Proof: $\Pi_B(n)$

⋮

Balance: \$50,
Proof: $\Pi_C(n)$

Motivation



Digest: d_n

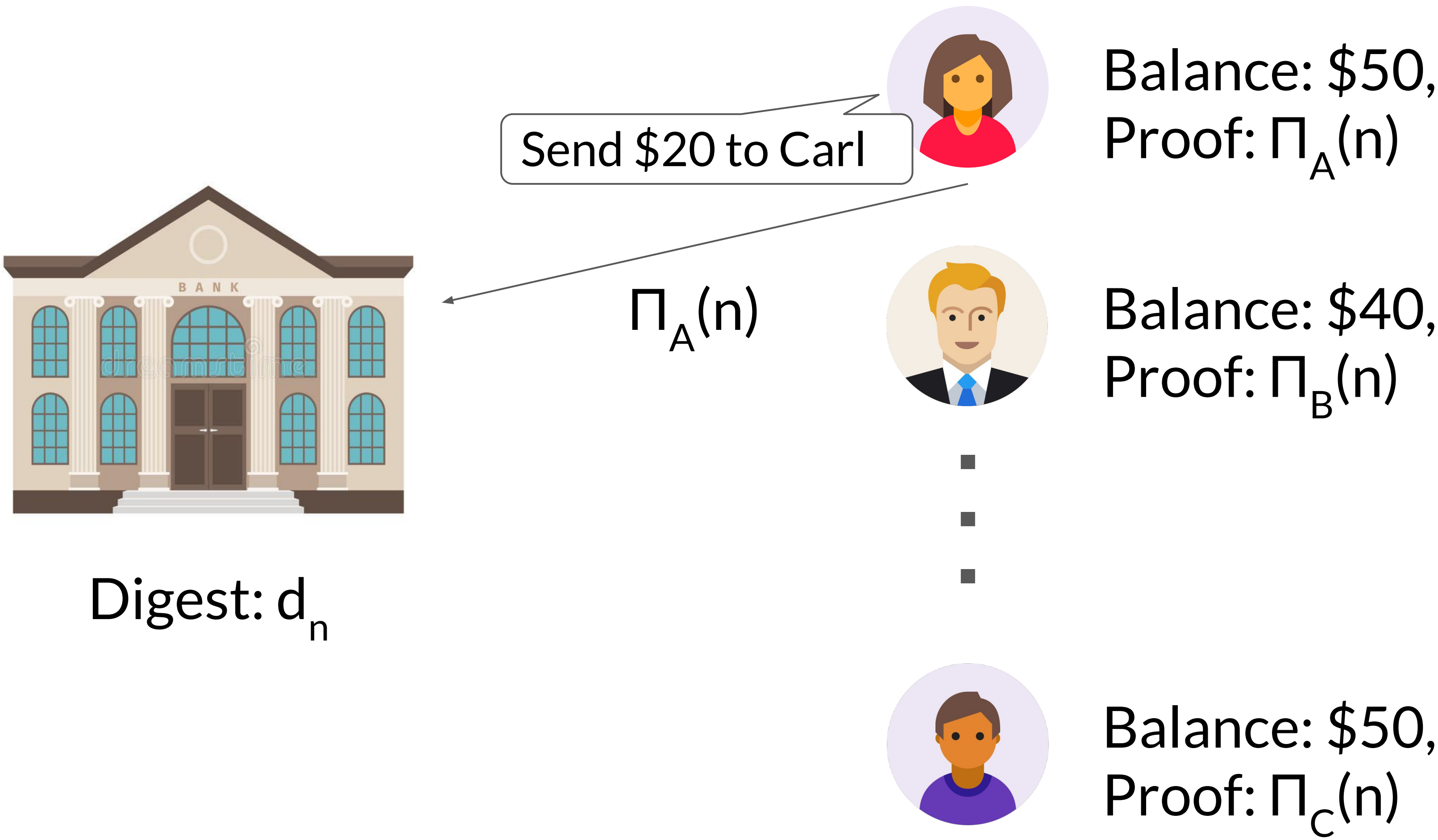
Send $20 to Carl

Balance: $50,
Proof: $\Pi_A(n)$

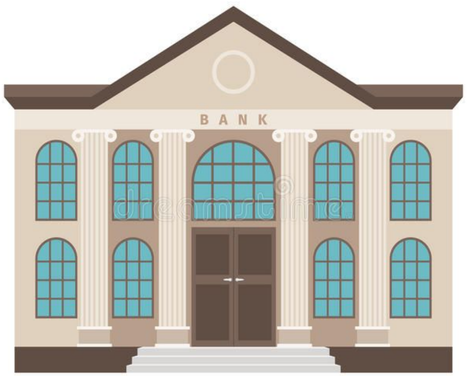
Balance: $40,
Proof: $\Pi_B(n)$

Balance: $50,
Proof: $\Pi_C(n)$

Motivation


Send $20 to Carl
$\Pi_A(n)$
Balance: $50,
Proof: $\Pi_A(n)$
Balance: $40,
Proof: $\Pi_B(n)$
Balance: $50,
Proof: $\Pi_C(n)$
BANK
Digest: d_n

Motivation



Digest: d_n

$Ver(d_n, A, \$20, \Pi_A(n))$

$\Pi_A(n)$

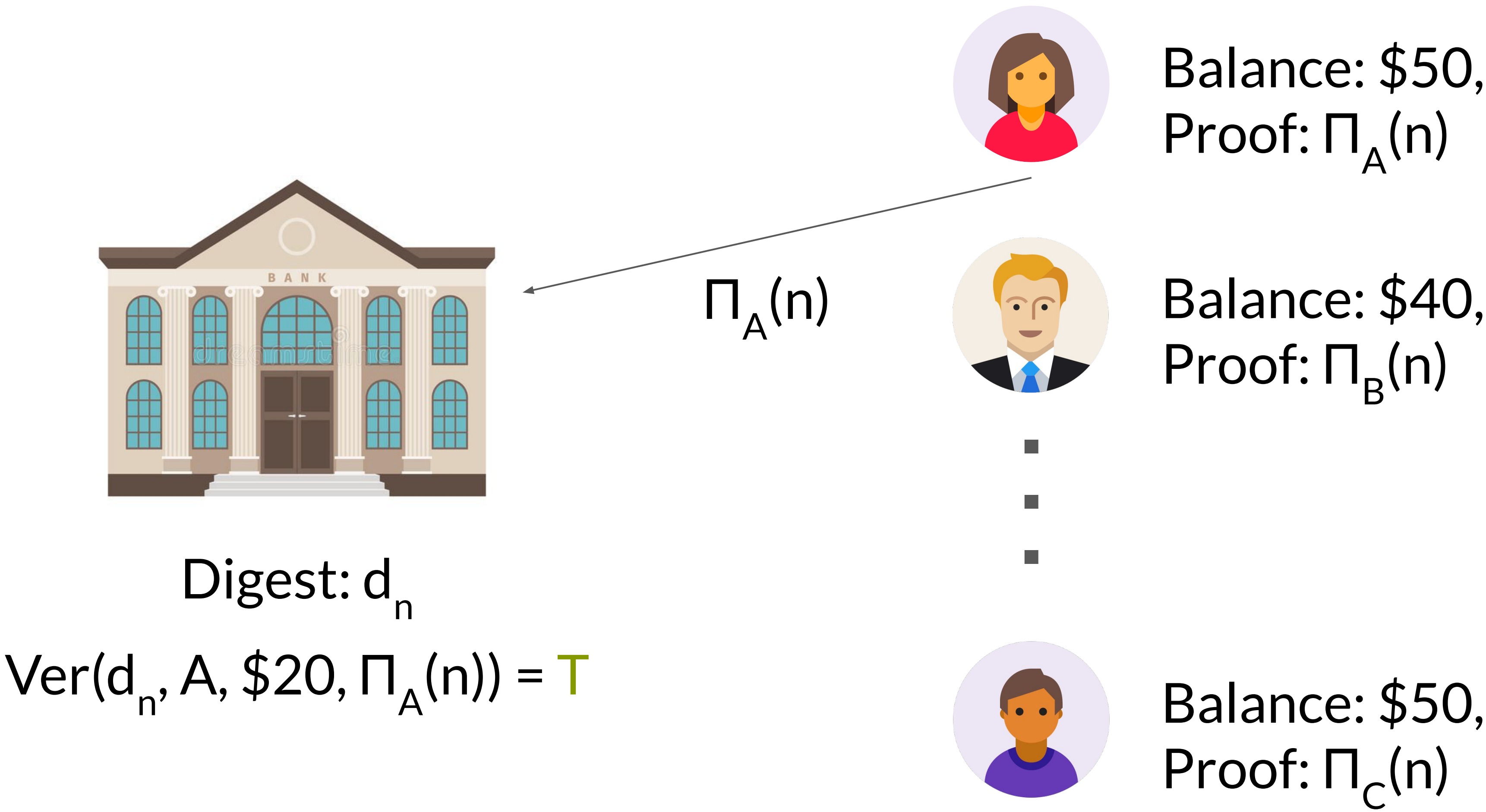
Balance: \$50,
Proof: $\Pi_A(n)$

Balance: \$40,
Proof: $\Pi_B(n)$

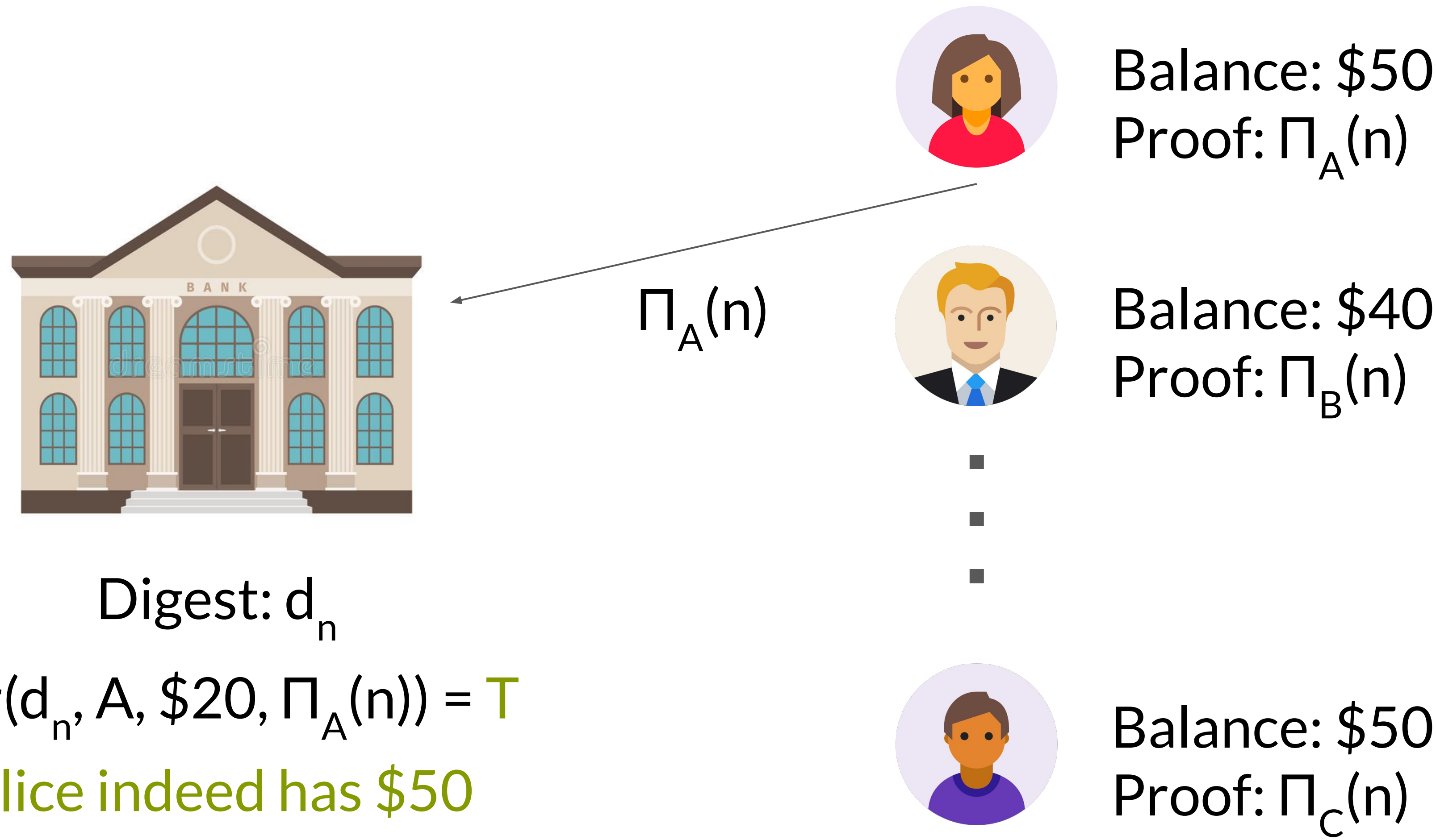
⋮

Balance: \$50,
Proof: $\Pi_C(n)$

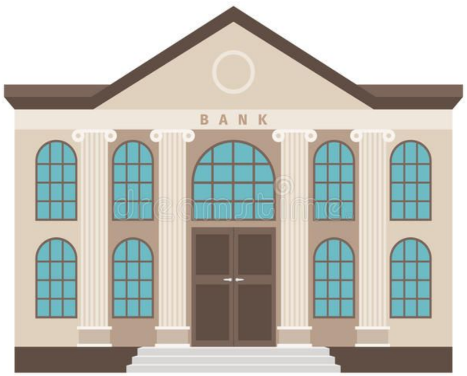
Motivation


Balance: \$50,
Proof: $\Pi_A(n)$
$\Pi_A(n)$
Balance: \$40,
Proof: $\Pi_B(n)$
Digest: d_n
$Ver(d_n, A, \$20, \Pi_A(n)) = T$
Balance: \$50,
Proof: $\Pi_C(n)$

Motivation


Balance: $50,
Proof: $\Pi_A(n)$
$\Pi_A(n)$
Balance: $40,
Proof: $\Pi_B(n)$
Balance: $50,
Proof: $\Pi_C(n)$
Digest: d_n
Ver(d_n, A, $20, $\Pi_A(n)$) = T
Alice indeed has $50

Motivation



Balance: \$50,
Proof: $\Pi_A(n)$

$\Pi_A(n)$

Balance: \$40,
Proof: $\Pi_B(n)$

Balance: \$50,
Proof: $\Pi_C(n)$

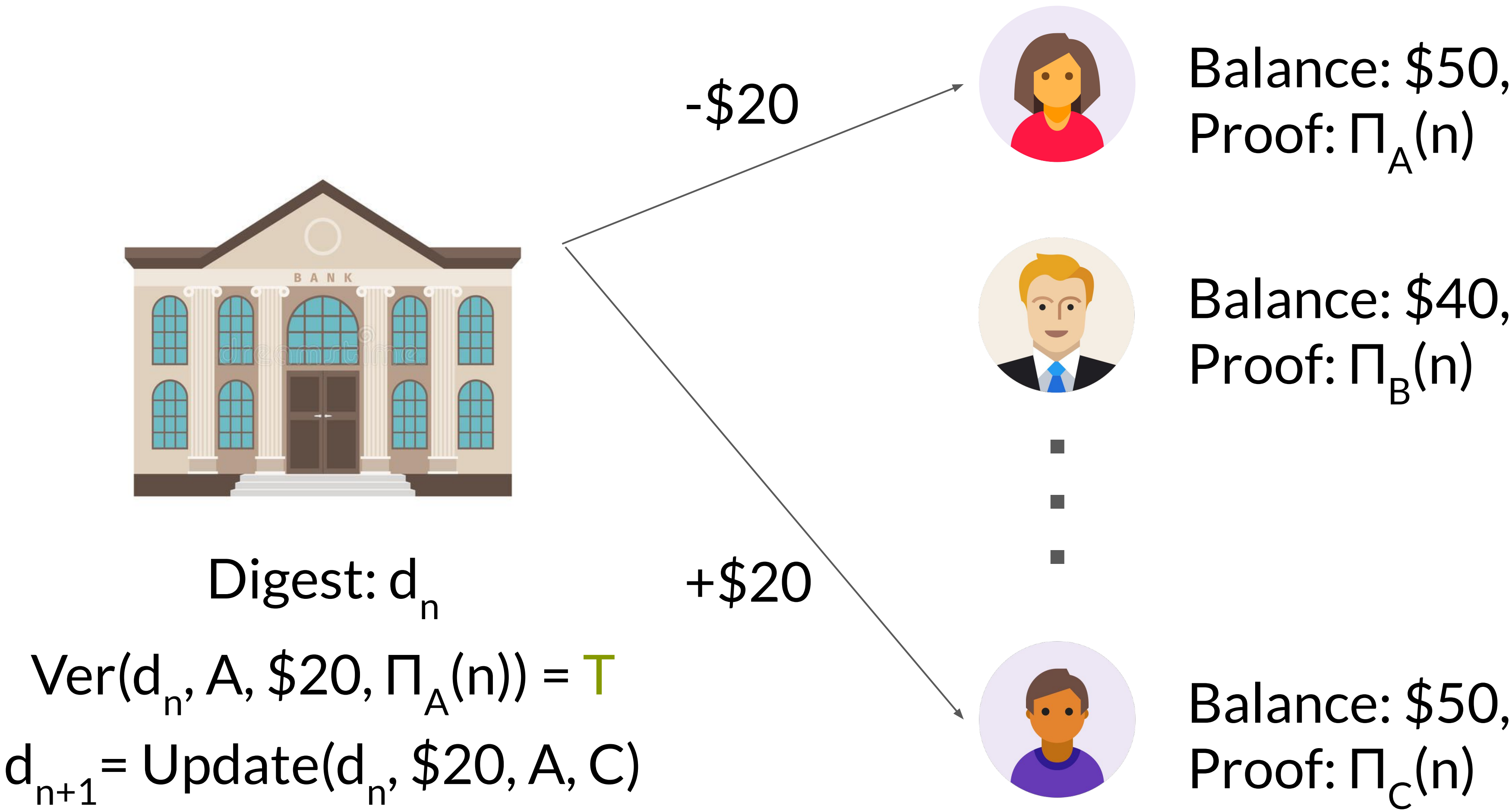
Digest: d_n

$\mathrm{Ver}(d_n, A, \$20, \Pi_A(n)) = T$

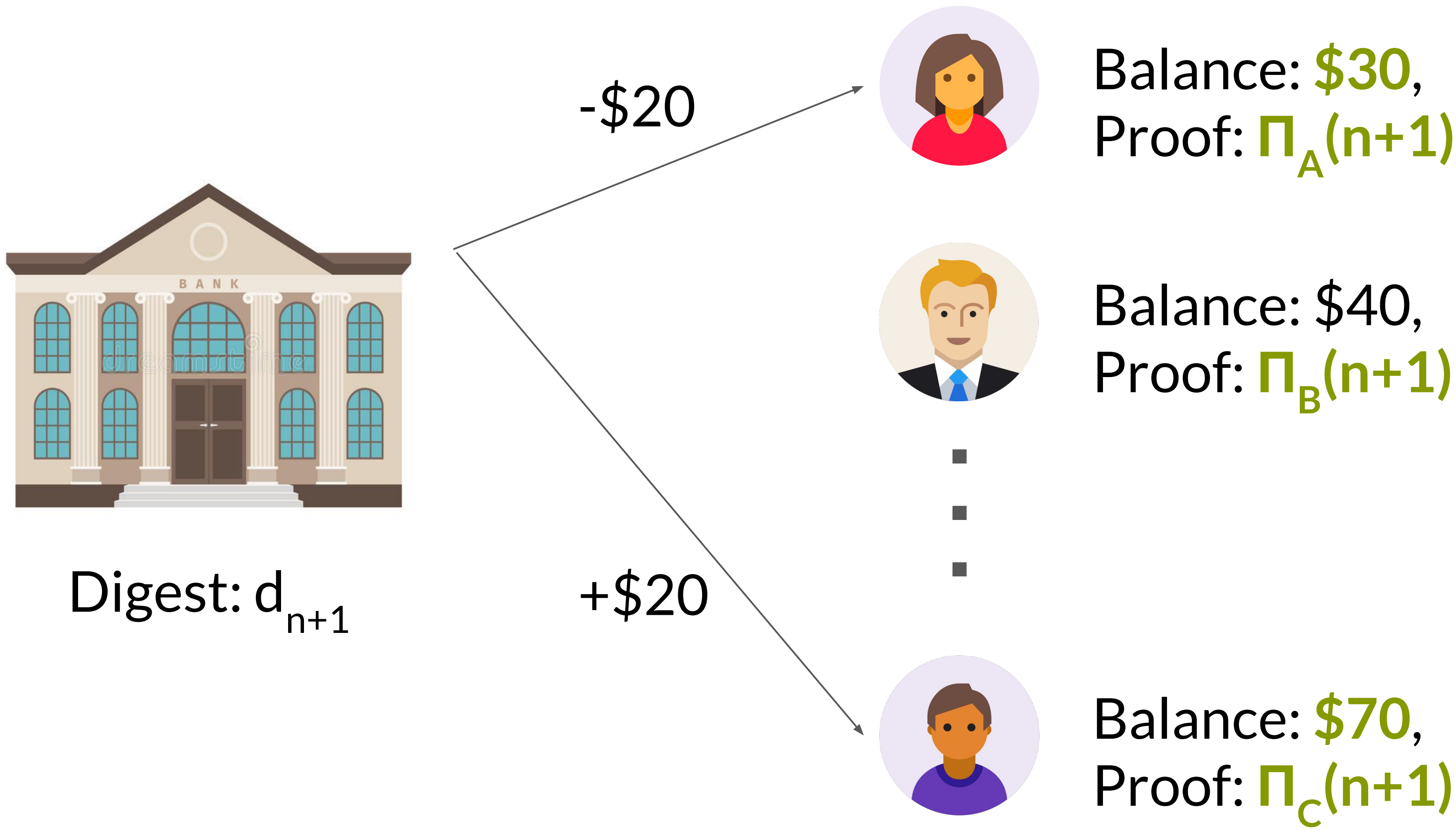
Alice indeed has \$50

The server only stores a 32-byte digest d_n

Motivation

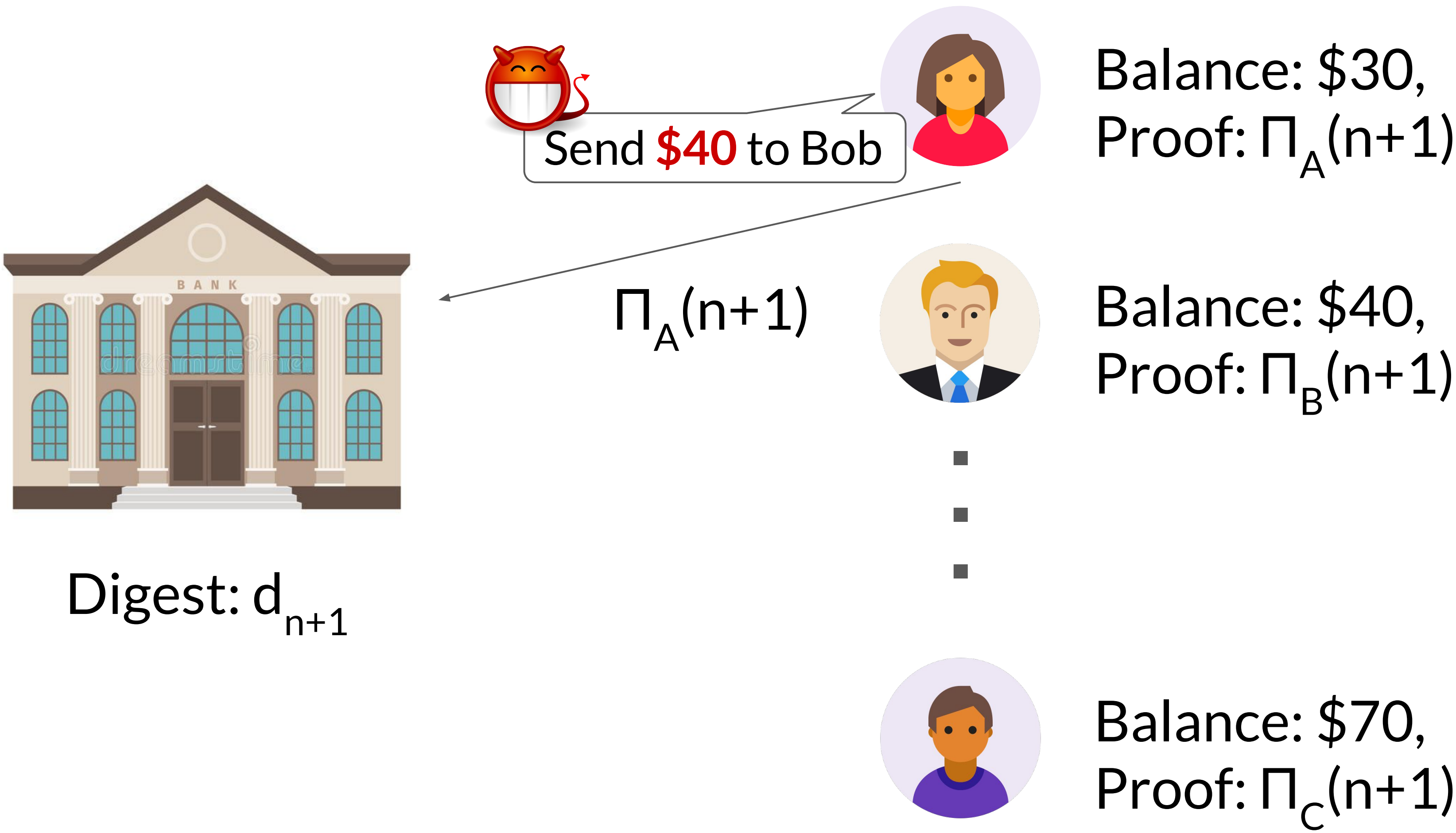

-$20
+$20
Digest: d_n
$Ver(d_n, A, \$20, \Pi_A(n)) = T$
$d_{n+1} = Update(d_n, \$20, A, C)$
Balance: $50,
Proof: $\Pi_A(n)$
Balance: $40,
Proof: $\Pi_B(n)$
Balance: $50,
Proof: $\Pi_C(n)$

Motivation



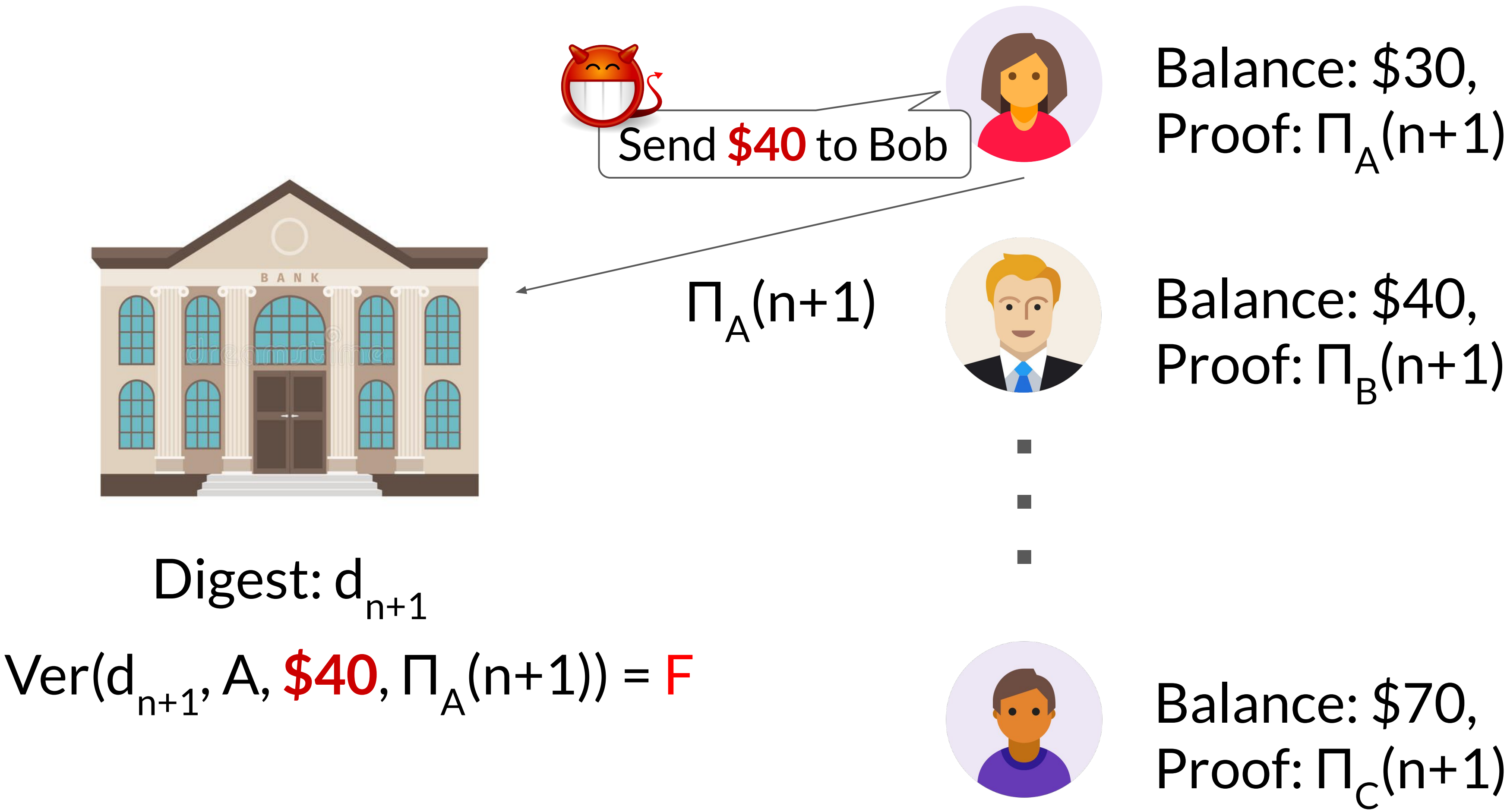
-$20
+$20
Digest: d_{n+1}
Balance: $30,
Proof: $\Pi_A(n+1)$
Balance: $40,
Proof: $\Pi_B(n+1)$
Balance: $70,
Proof: $\Pi_C(n+1)$

Motivation



Send $40 to Bob
Balance: $30,
Proof: $\Pi_A(n+1)$
$\Pi_A(n+1)$
BANK
Balance: $40,
Proof: $\Pi_B(n+1)$
Balance: $70,
Proof: $\Pi_C(n+1)$
Digest: d_{n+1}

Motivation


Send $40 to Bob
Balance: $30,
Proof: Π_A(n+1)
BANK
Π_A(n+1)
Balance: $40,
Proof: Π_B(n+1)
Balance: $70,
Proof: Π_C(n+1)
Digest: d_{n+1}
Ver(d_{n+1}, A, $40, Π_A(n+1)) = F

Motivation

Send \$40 to Bob

Balance: \$30,
Proof: $\Pi_A(n+1)$

$\Pi_A(n+1)$

Balance: \$40,
Proof: $\Pi_B(n+1)$

Balance: \$70,
Proof: $\Pi_C(n+1)$

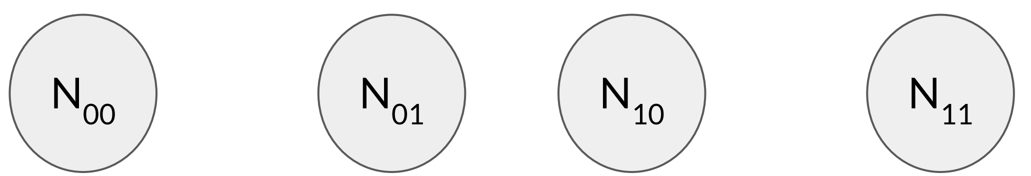
Digest: d_{n+1}

$Ver(d_{n+1}, A, \mathbf{\$40}, \Pi_A(n+1)) = F$

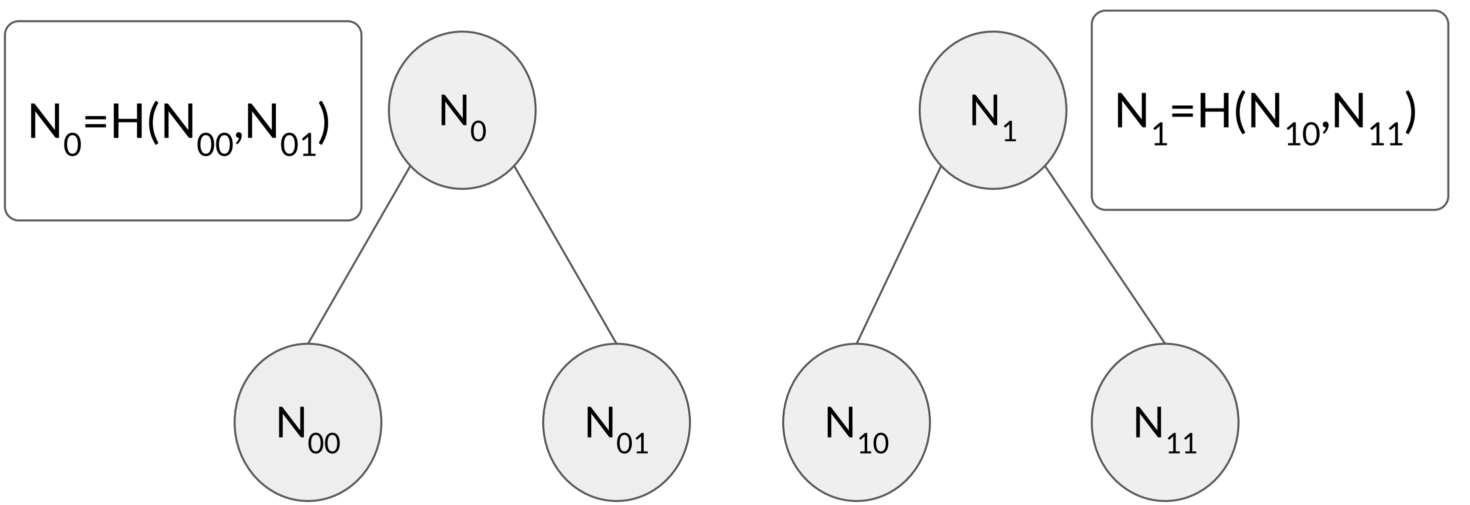
Alice does not have sufficient funds

How can we do this?

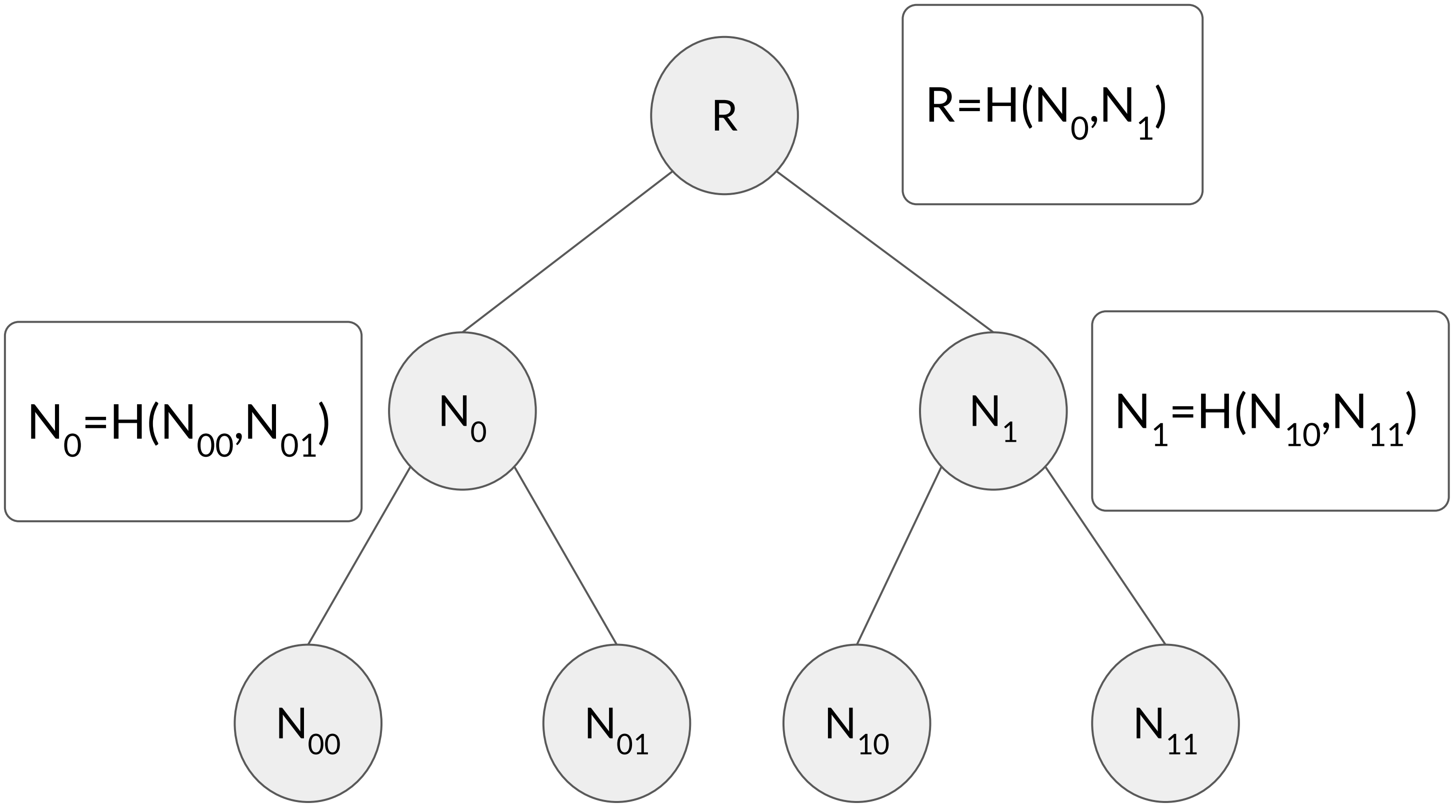
How can we do this? Merkle Hash Trees (MHT)!



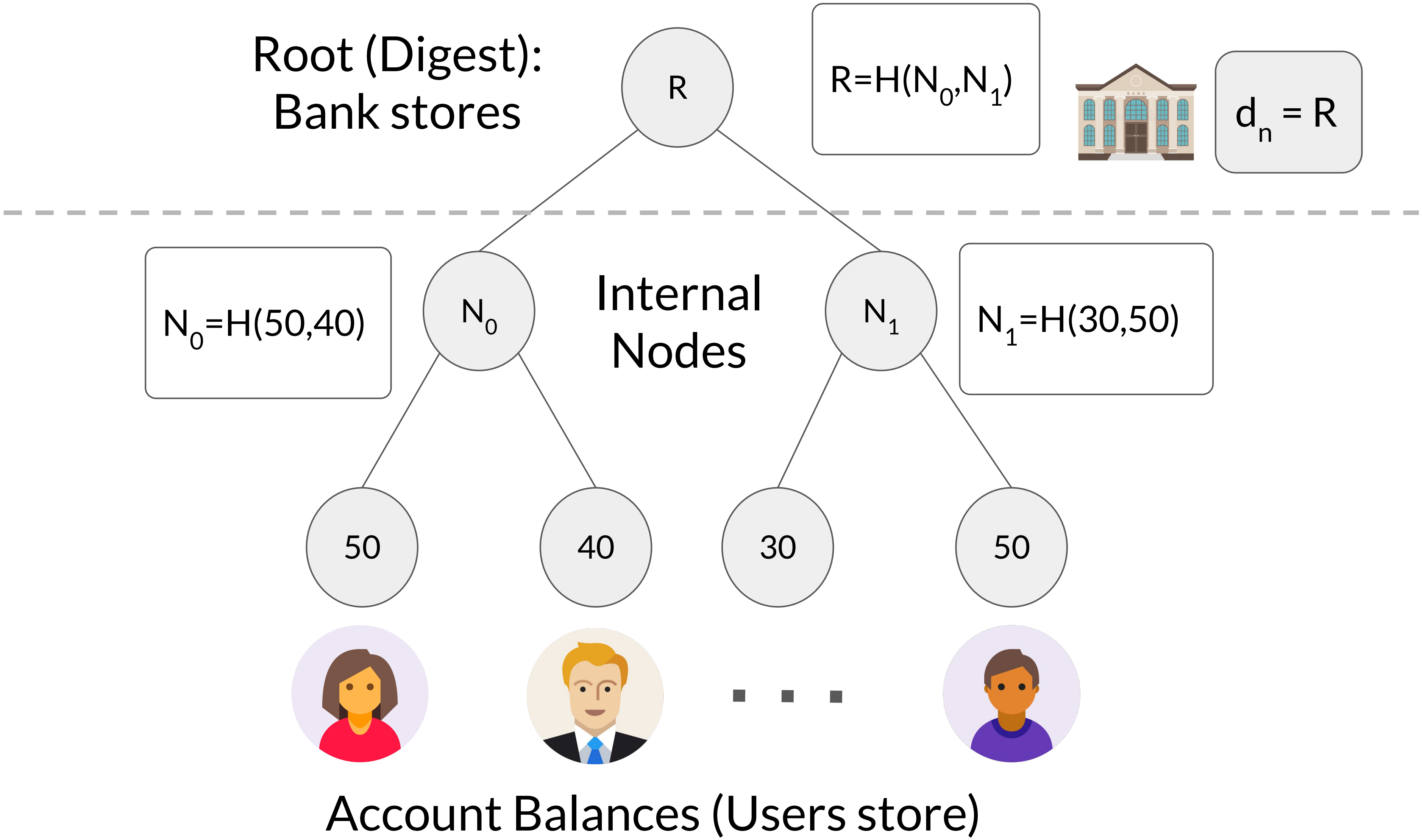
Building an MHT



Building an MHT

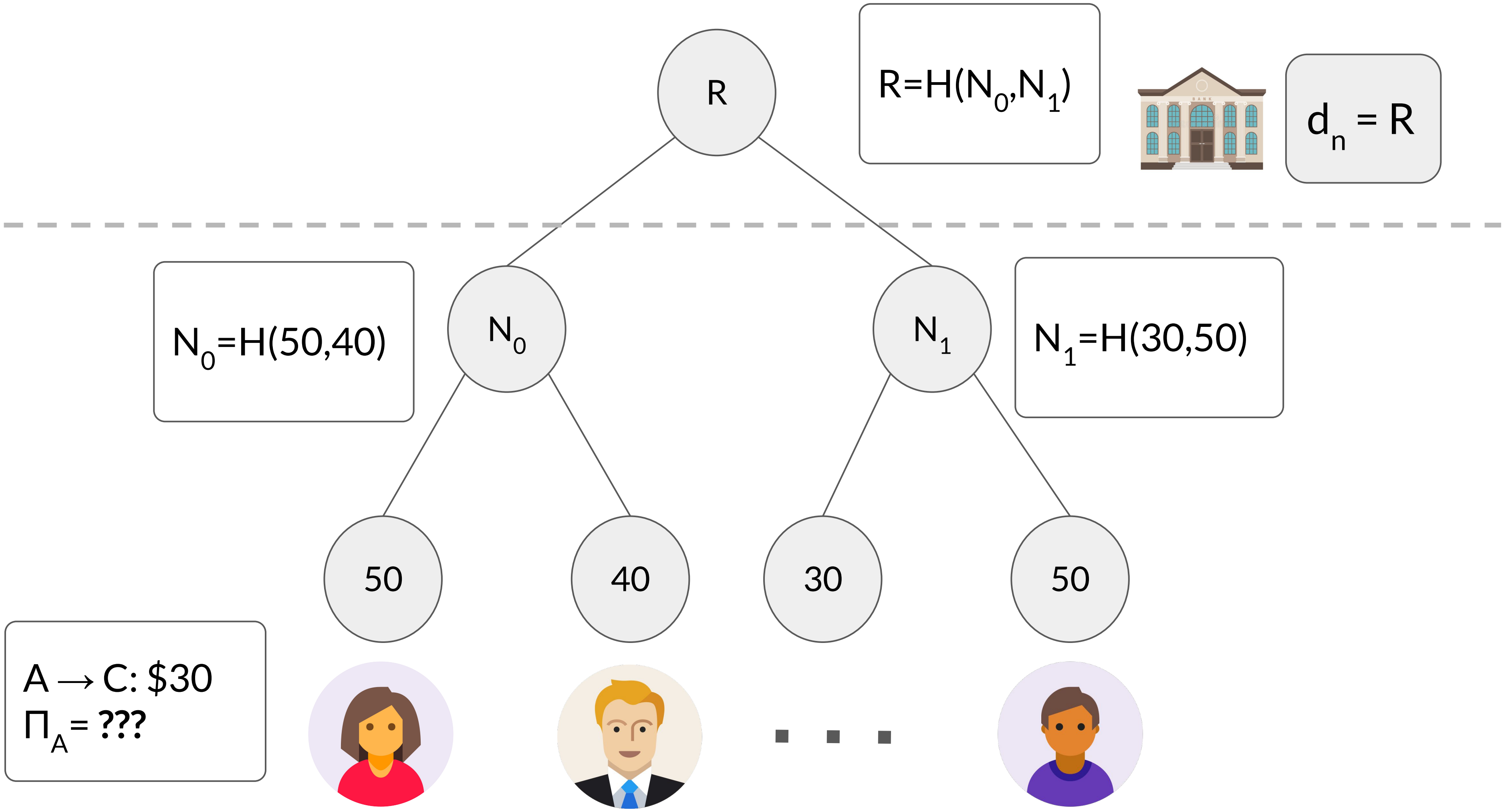


MHT Example

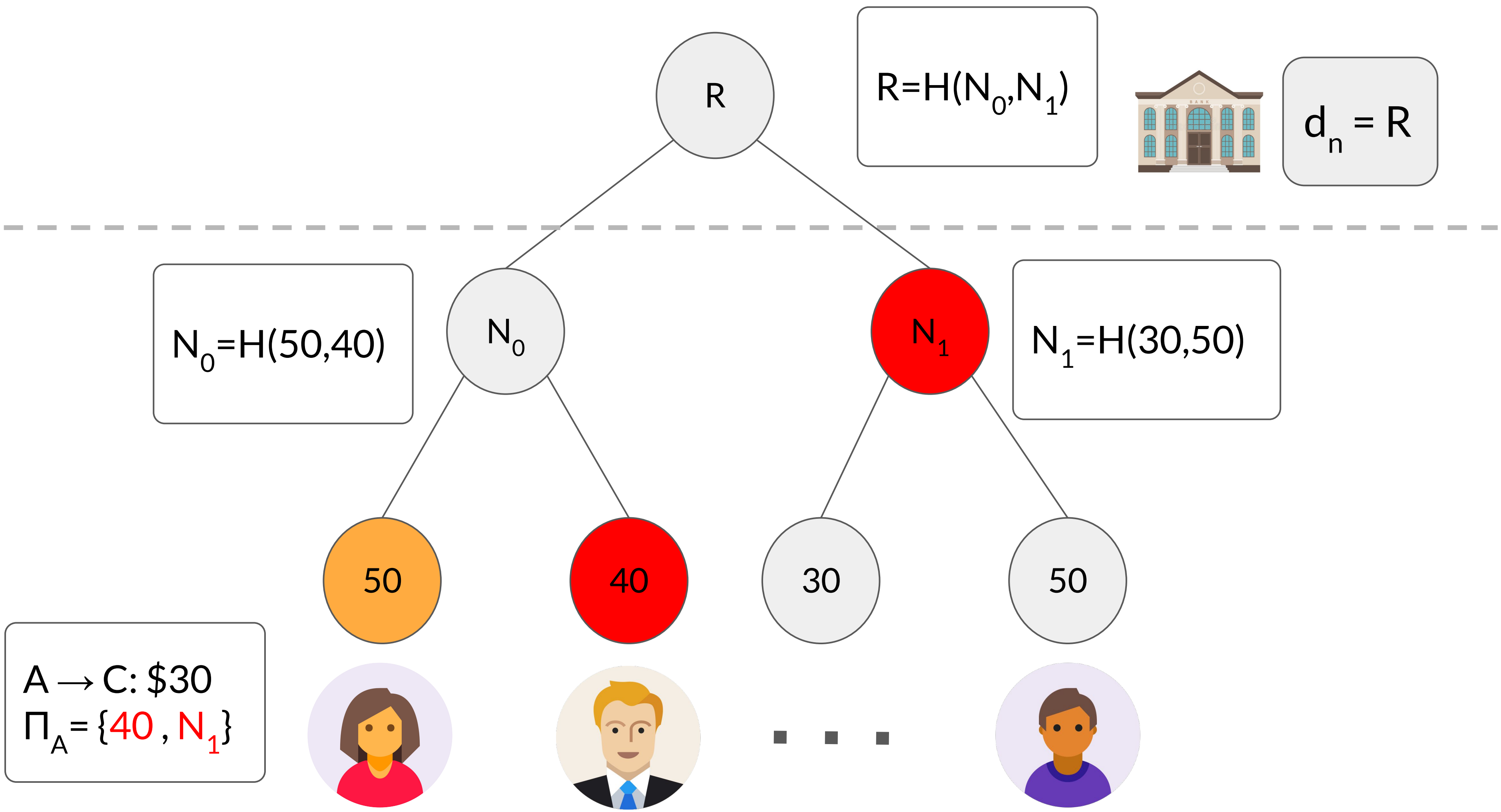


Root (Digest):
Bank stores
R
R=H(N0,N1)
dn = R
N0=H(50,40)
N0
Internal
Nodes
N1
N1=H(30,50)
50
40
30
50
Account Balances (Users store)

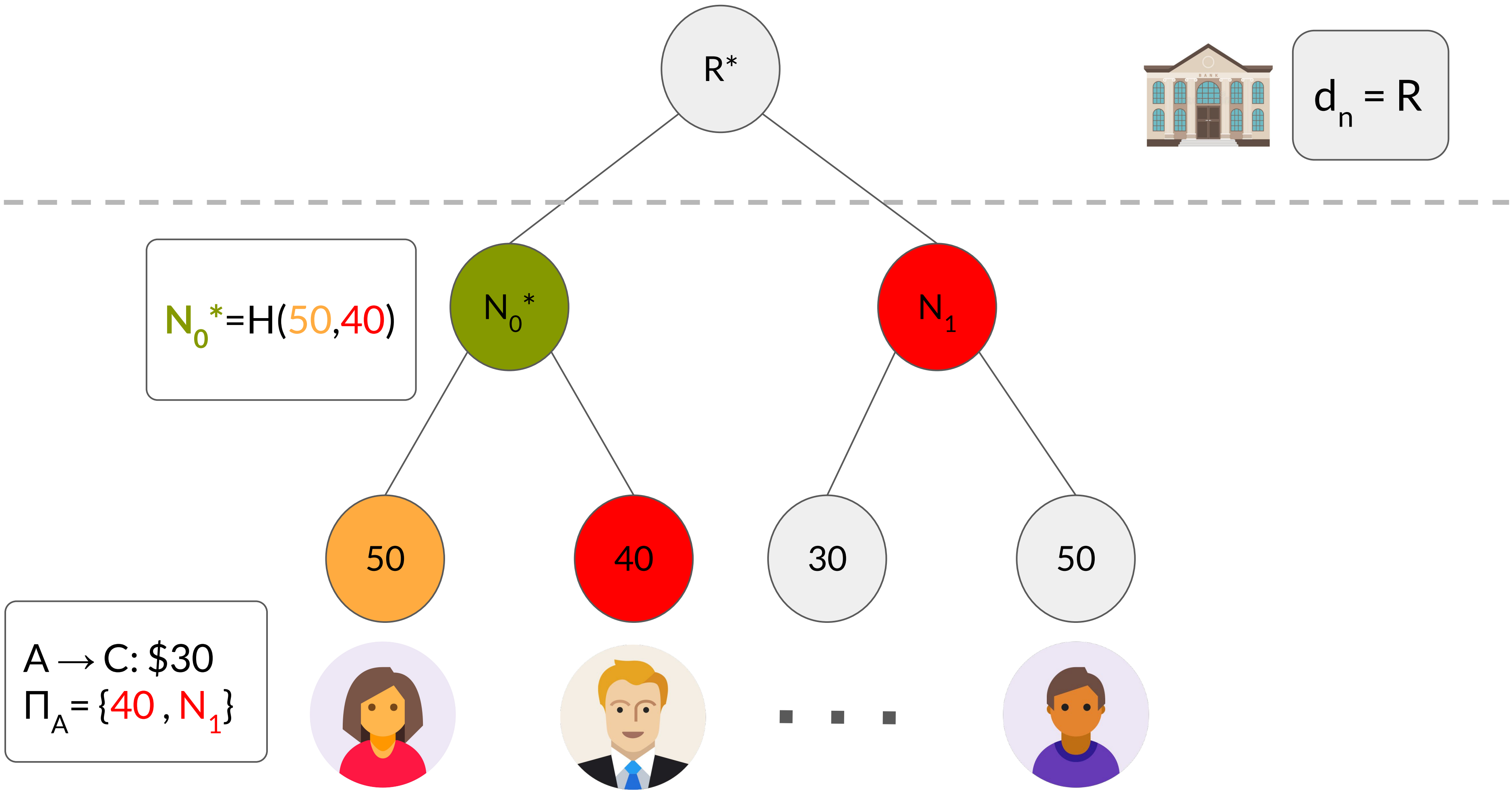
MHT Proof of Balance



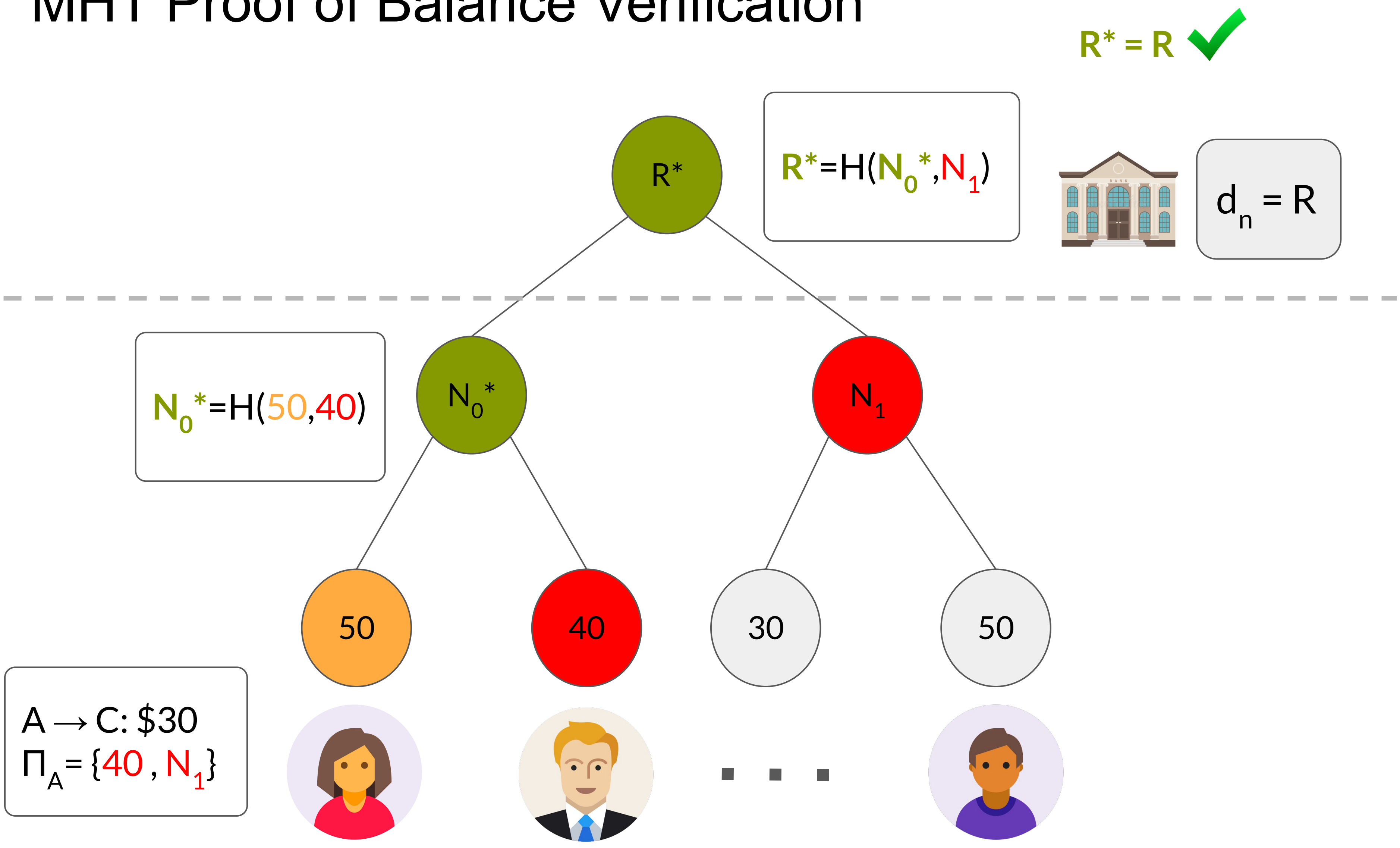
MHT Proof of Balance



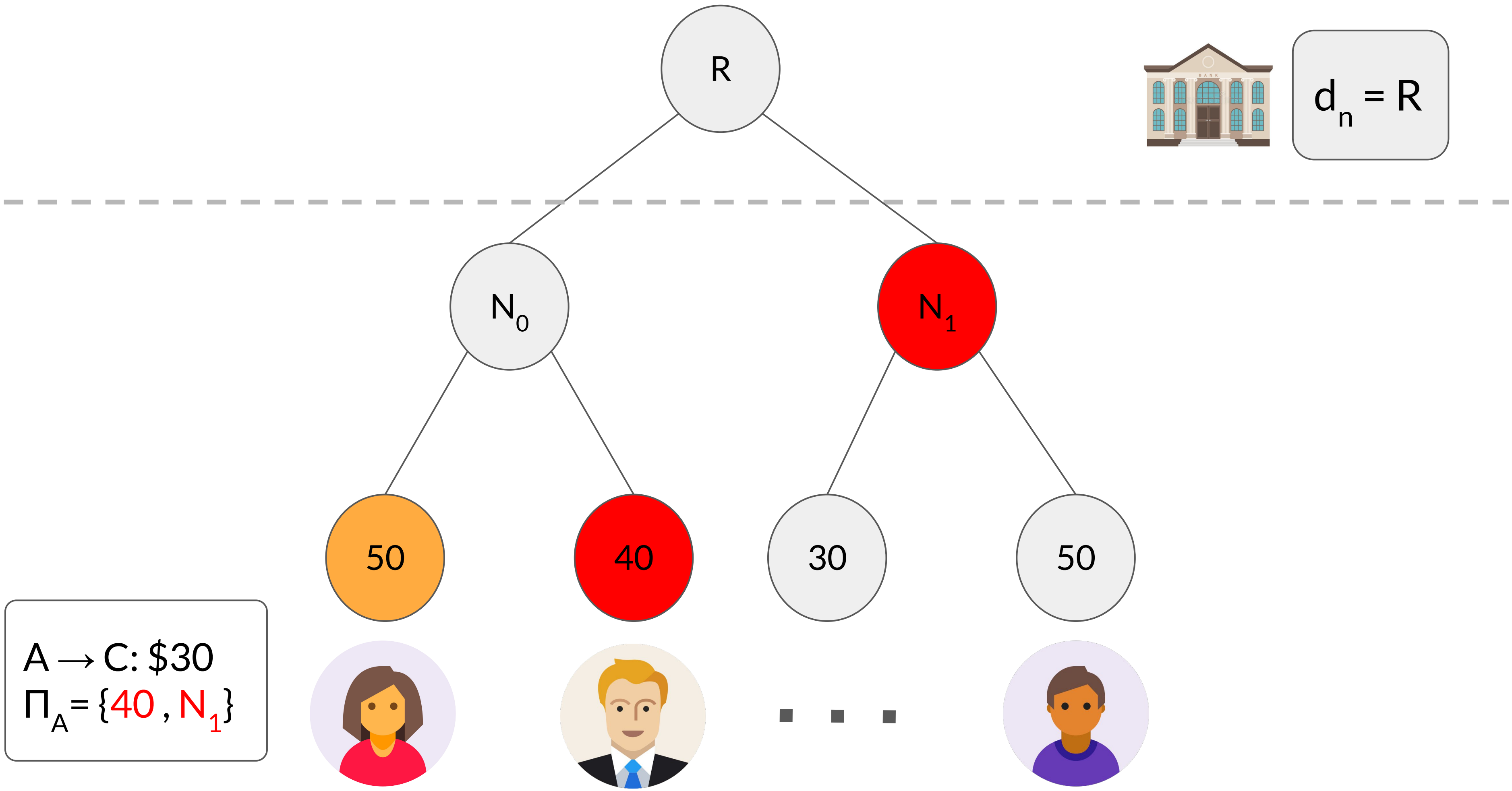
MHT Proof of Balance Verification



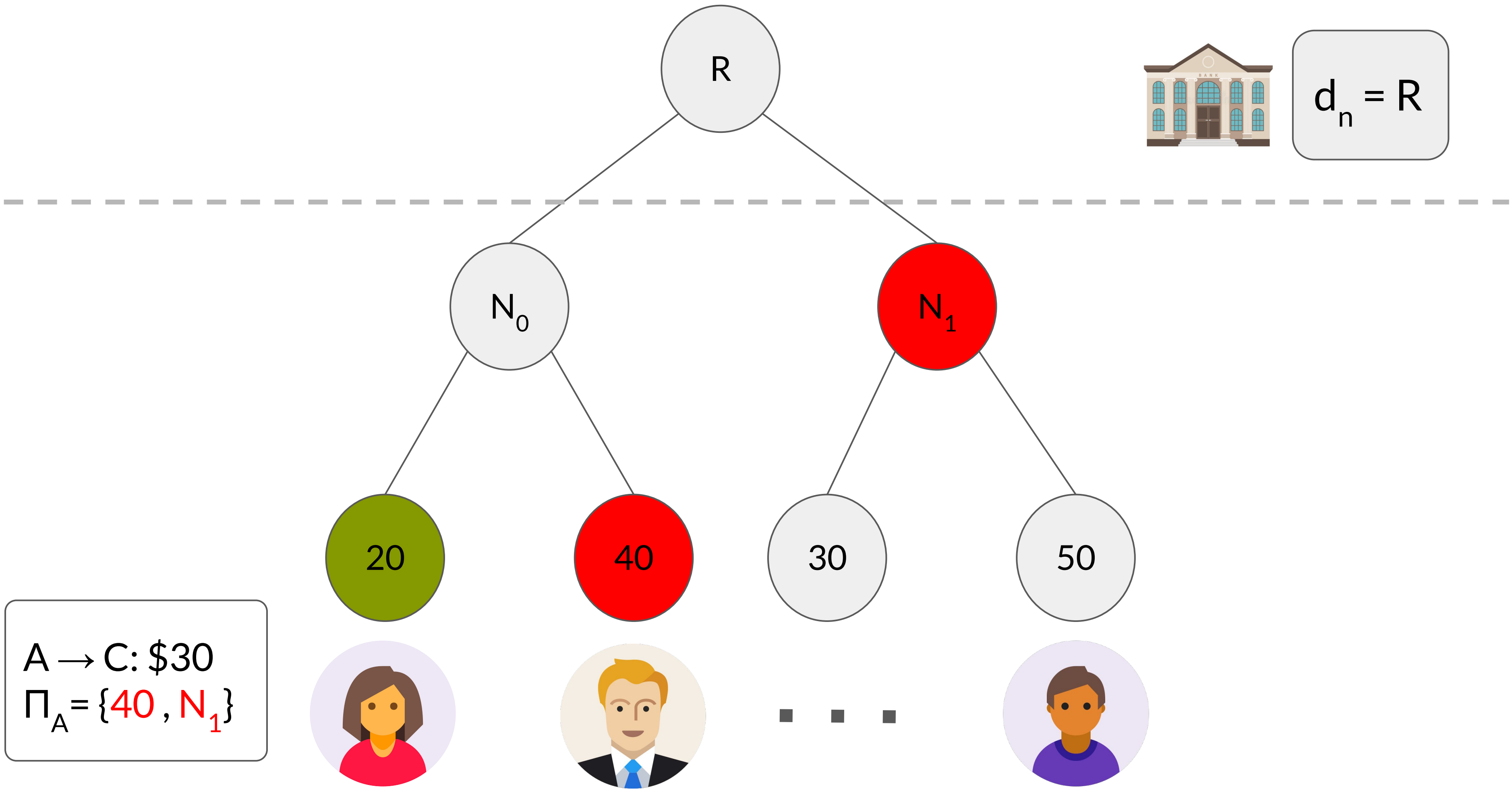
MHT Proof of Balance Verification



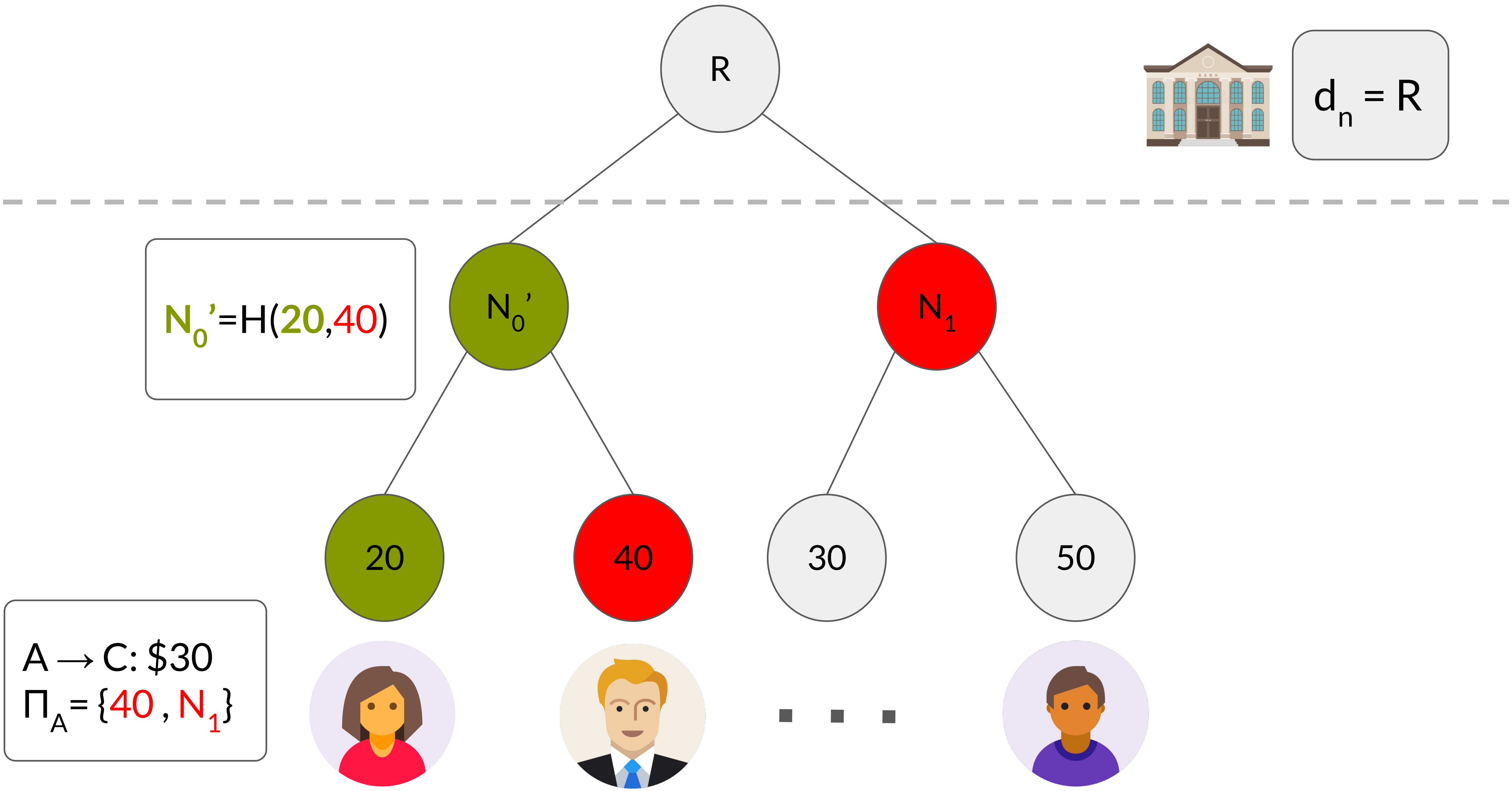
MHT Updating Balance



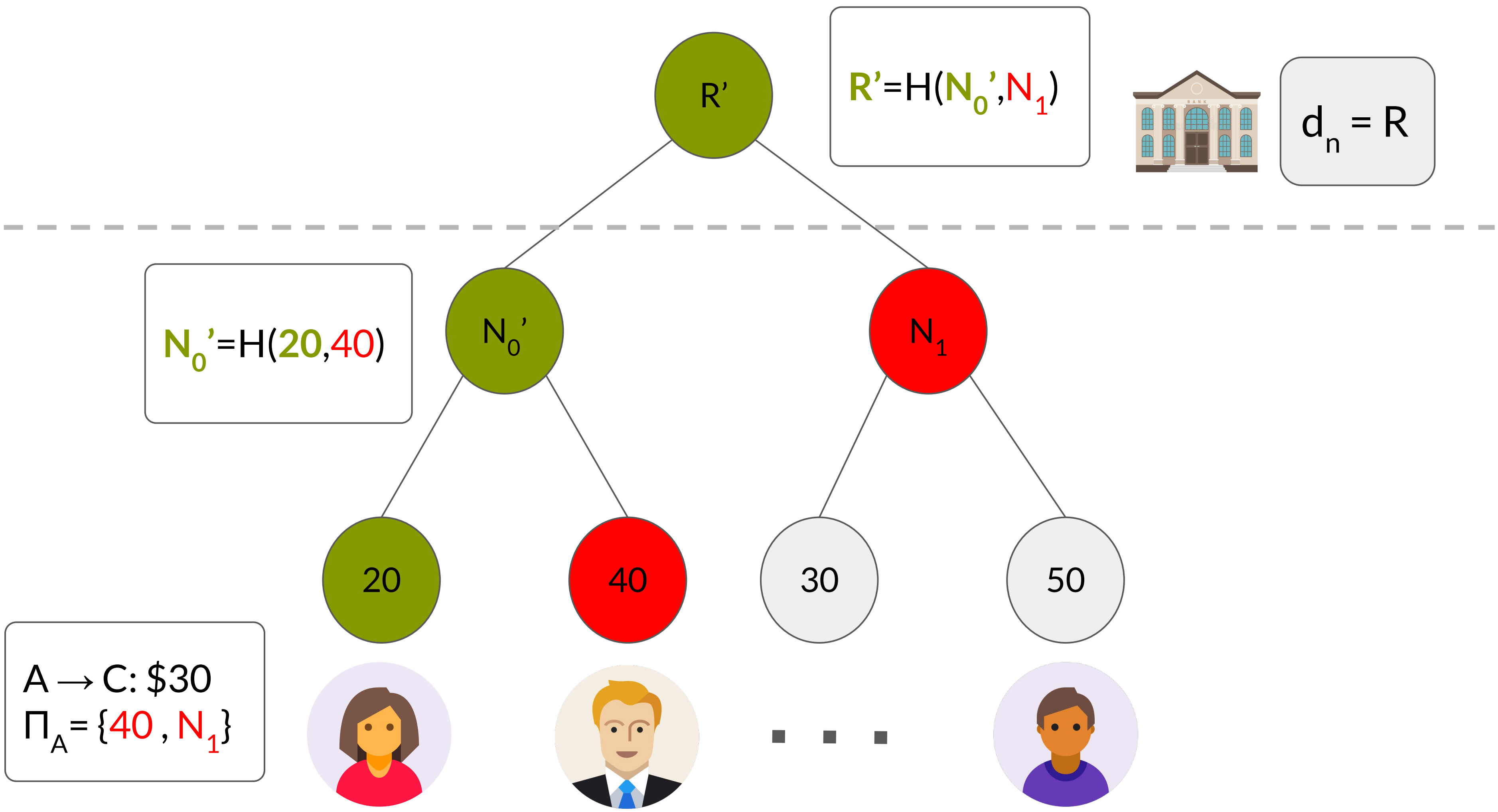
MHT Updating Balance



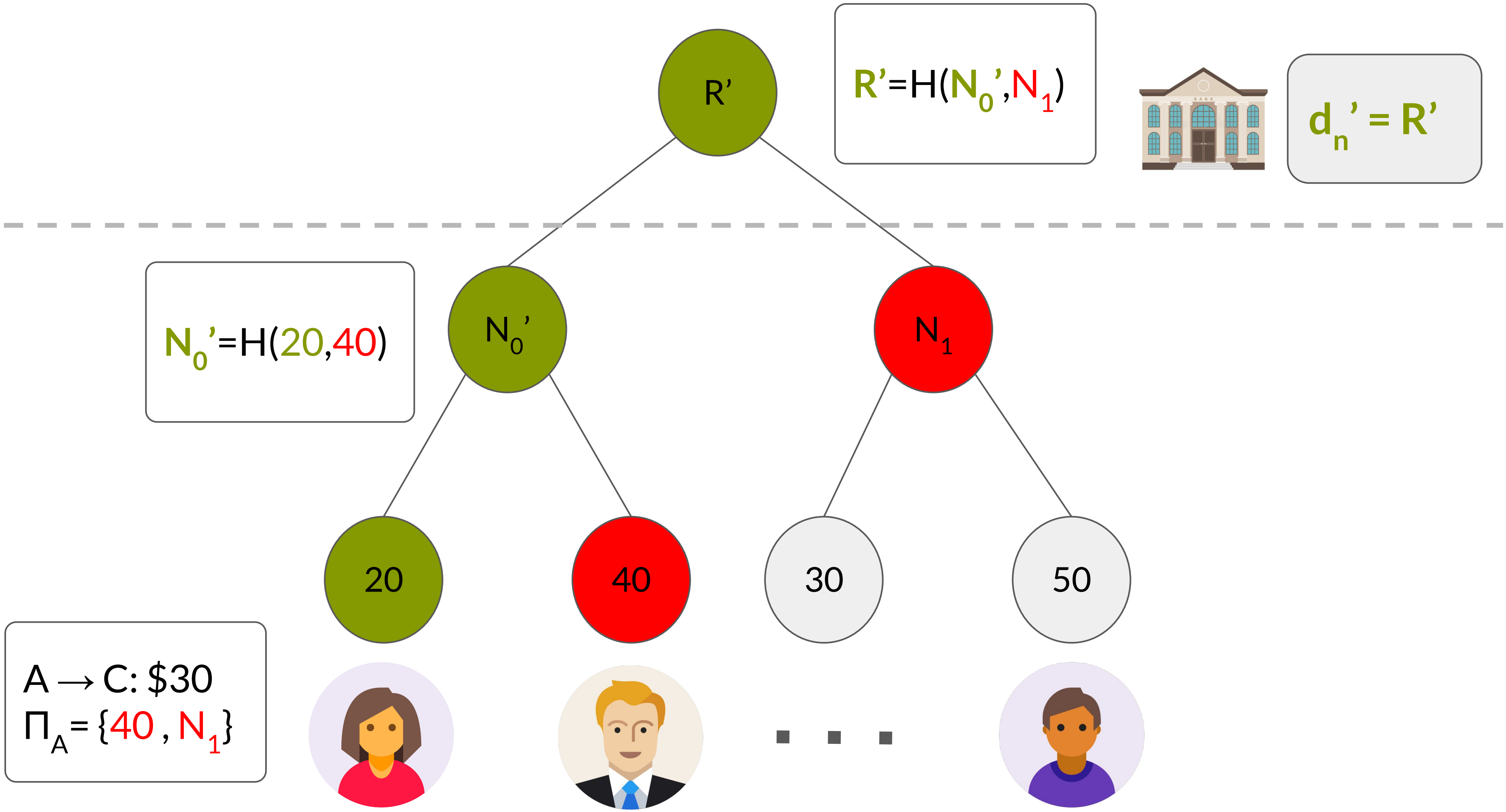
MHT Updating Balance



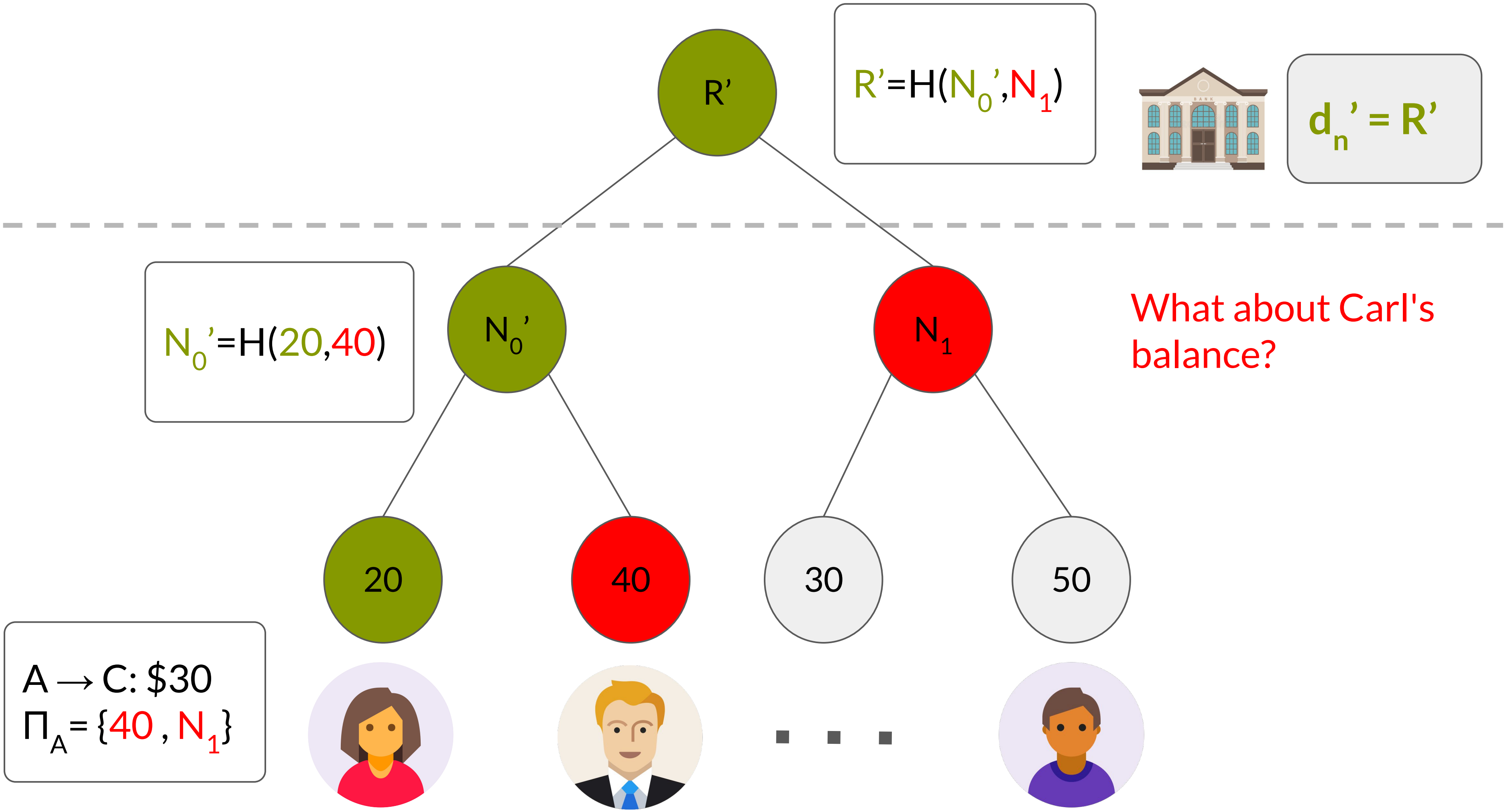
MHT Updating Balance



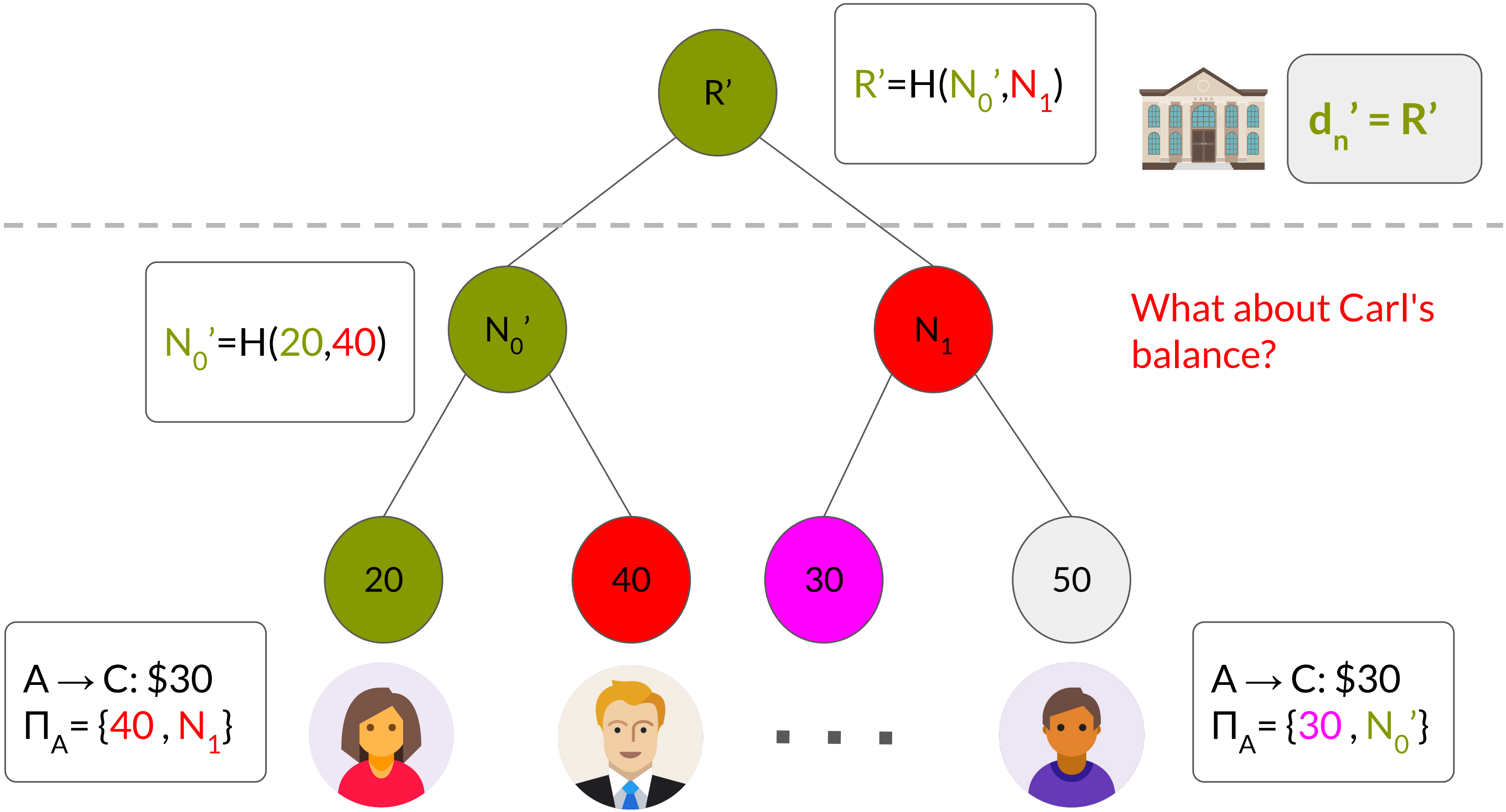
MHT Updating Balance



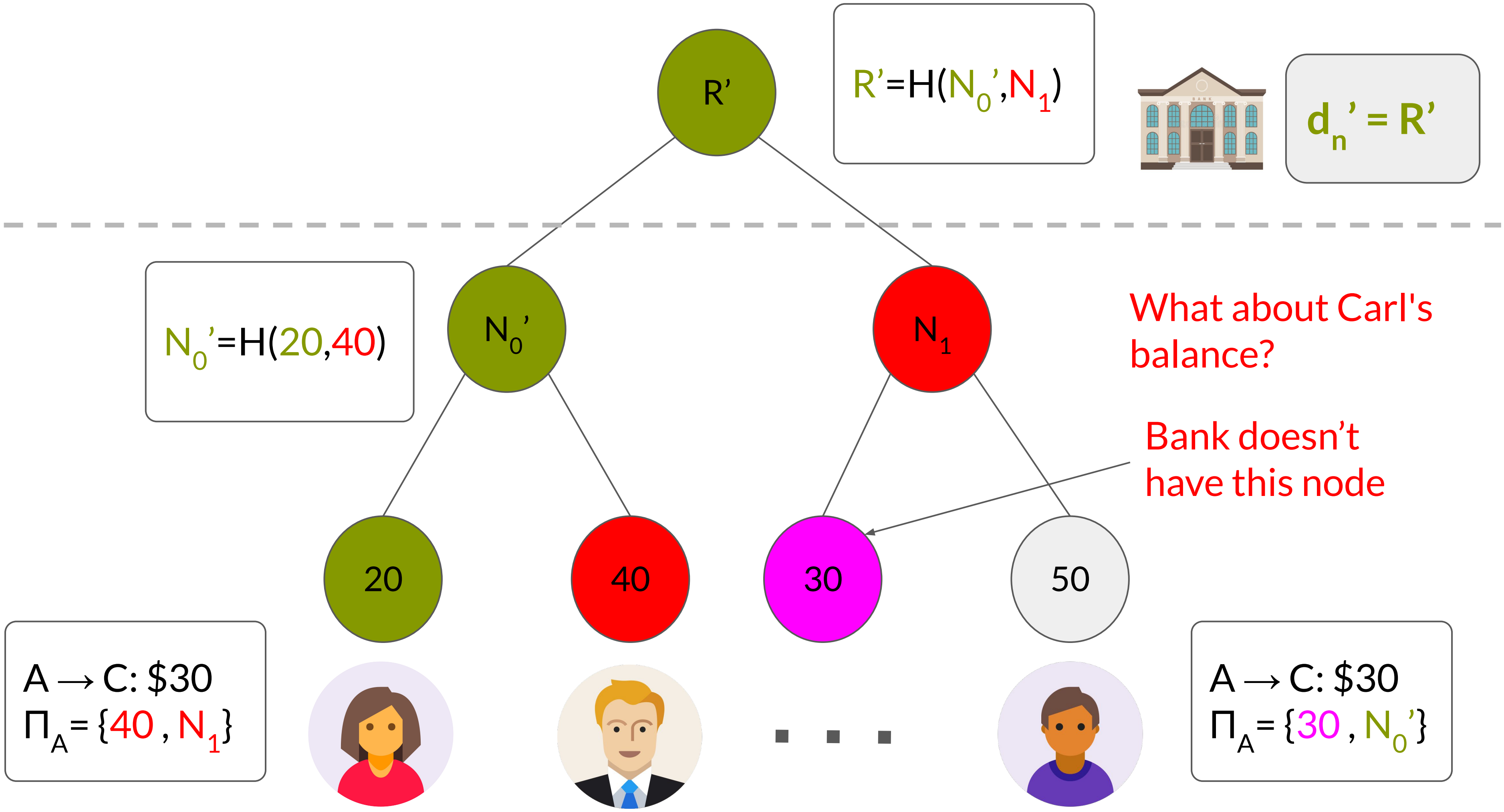
MHT Updating Balance



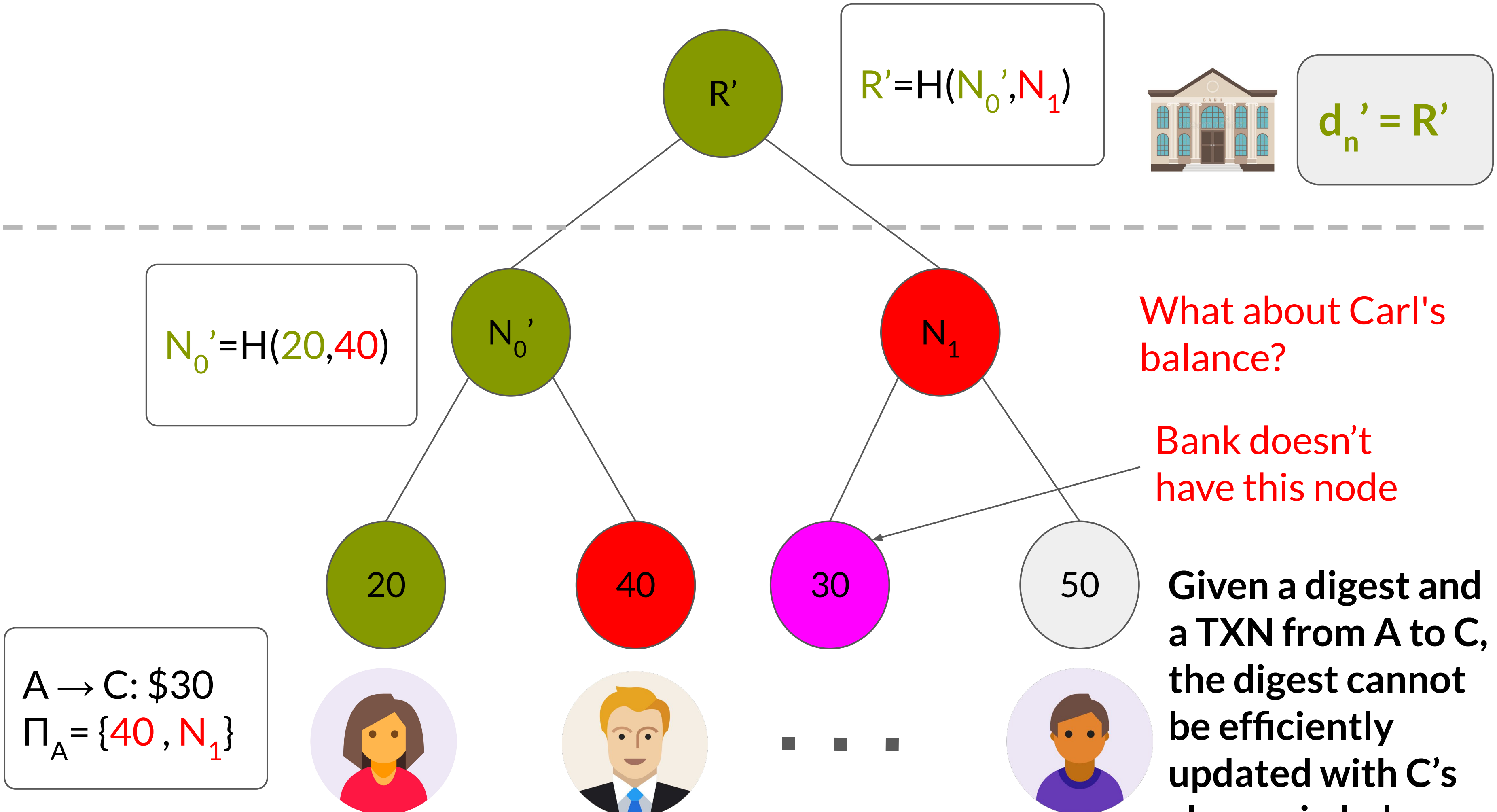
MHT Updating Sender’s Balance



MHT Updating Sender's Balance

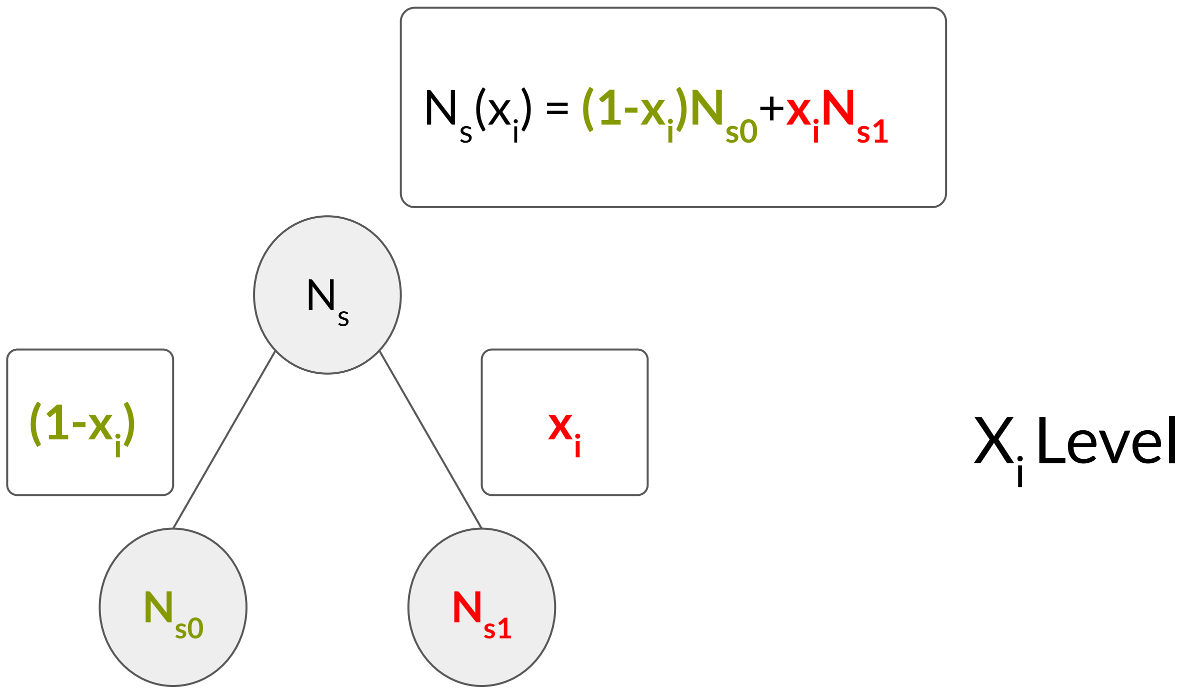


MHT Updating Sender's Balance

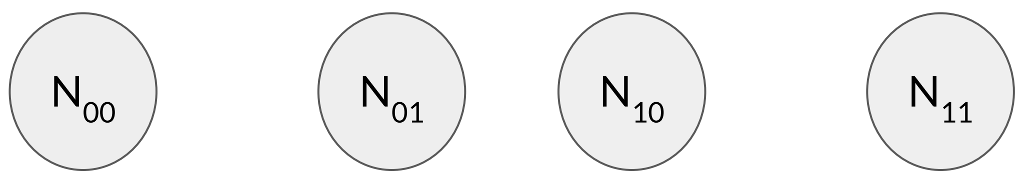


Given a digest and a TXN from A to C, the digest cannot be efficiently updated with C's change in balance

Building a Multivariate Polynonial Hash Tree (MPHT)

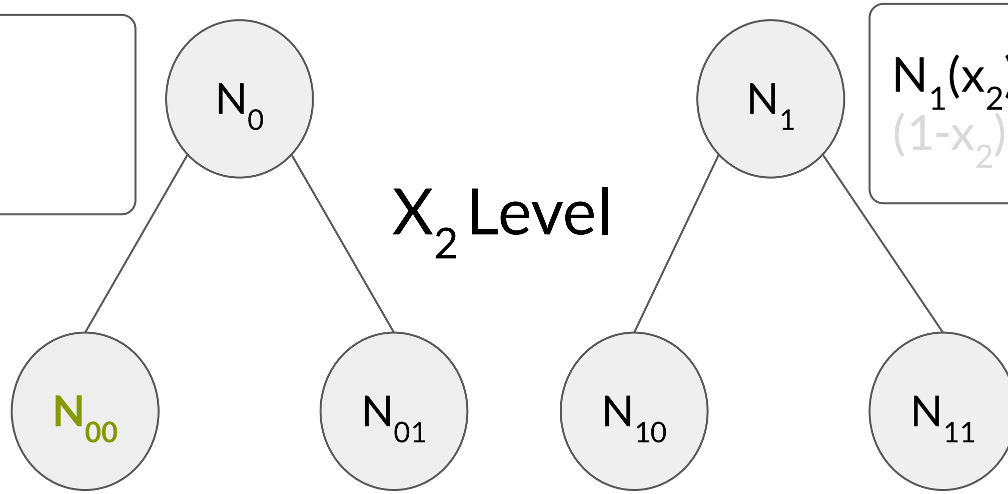


Building an MPHT



Building an MPHT

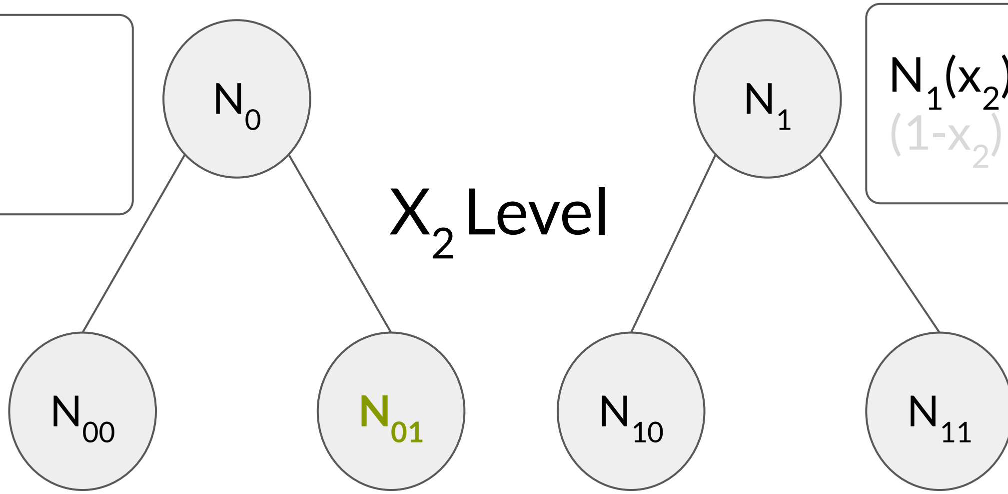
$N_0(x_2) =$
$(1-x_2)N_{00}+x_2N_{01}$



$N_1(x_2) =$
$(1-x_2)N_{10}+x_2N_{11}$

Building an MPHT

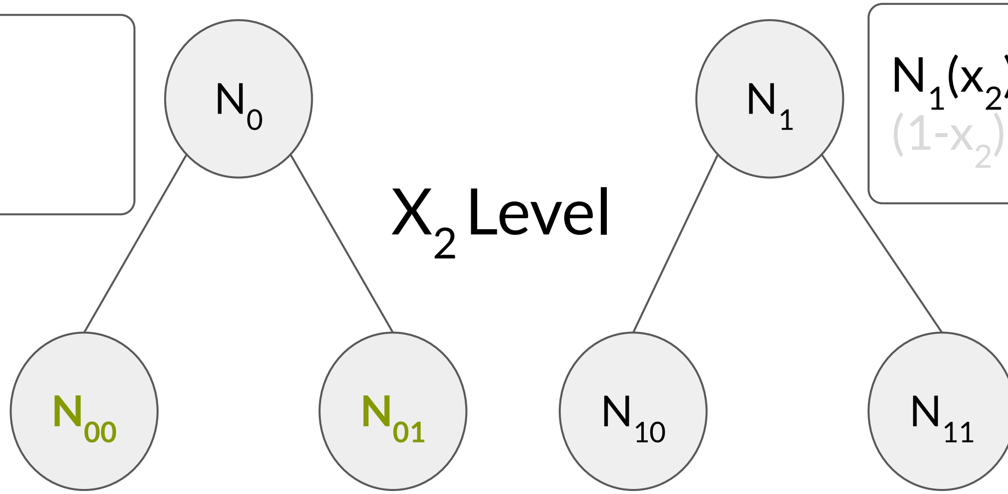
$N_0(x_2) = (1-x_2)N_{00} + x_2N_{01}$



$N_1(x_2) = (1-x_2)N_{10} + x_2N_{11}$

Building an MPHT

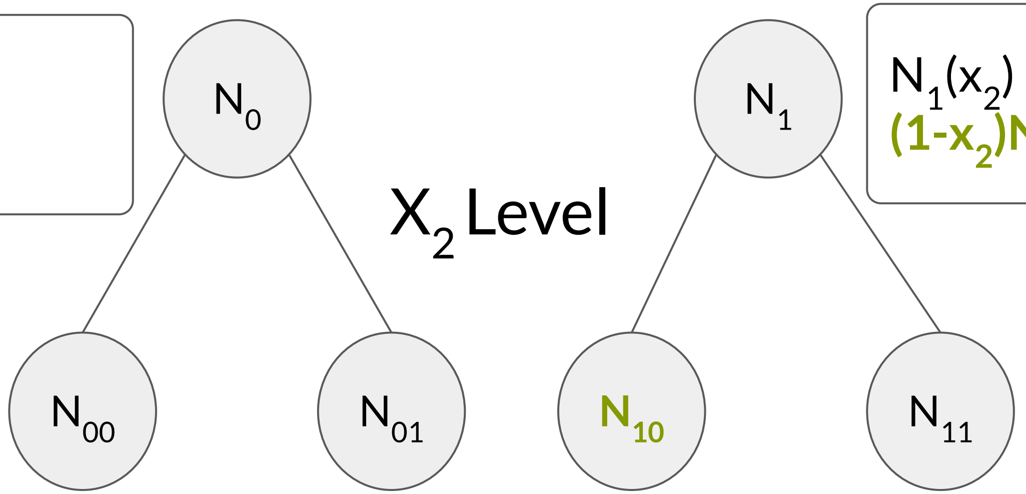
$N_0(x_2) = (1-x_2)N_{00}+x_2N_{01}$



$N_1(x_2) = (1-x_2)N_{10}+x_2N_{11}$

Building an MPHT

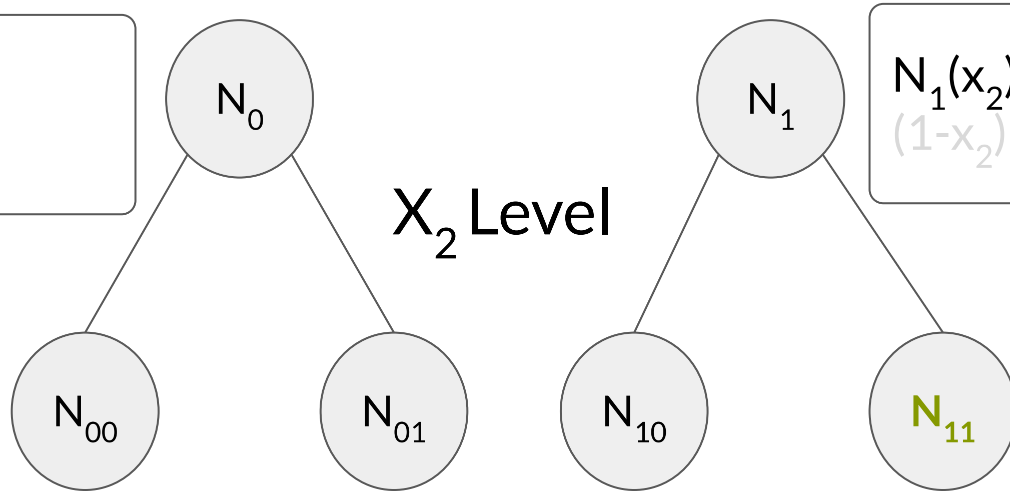
$N_0(x_2) =$
$(1-x_2)N_{00}+x_2N_{01}$



$N_1(x_2) =$
$(1-x_2)N_{10}+x_2N_{11}$

Building an MPHT

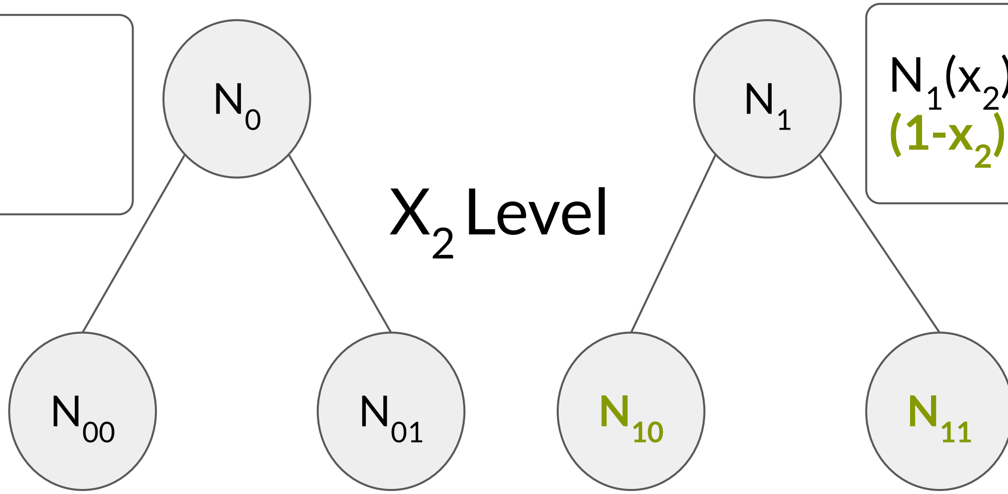
$N_0(x_2) = (1-x_2)N_{00}+x_2N_{01}$



$N_1(x_2) = (1-x_2)N_{10}+\mathbf{x_2N_{11}}$

Building an MPHT

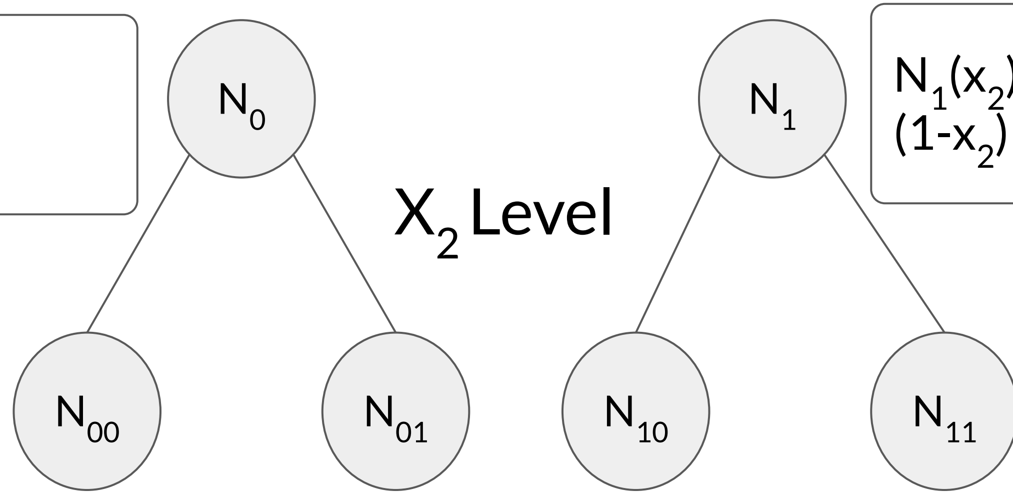
$N_0(x_2) =$
$(1-x_2)N_{00}+x_2N_{01}$



$N_1(x_2) =$
$(1-x_2)N_{10}+x_2N_{11}$

Building an MPHT

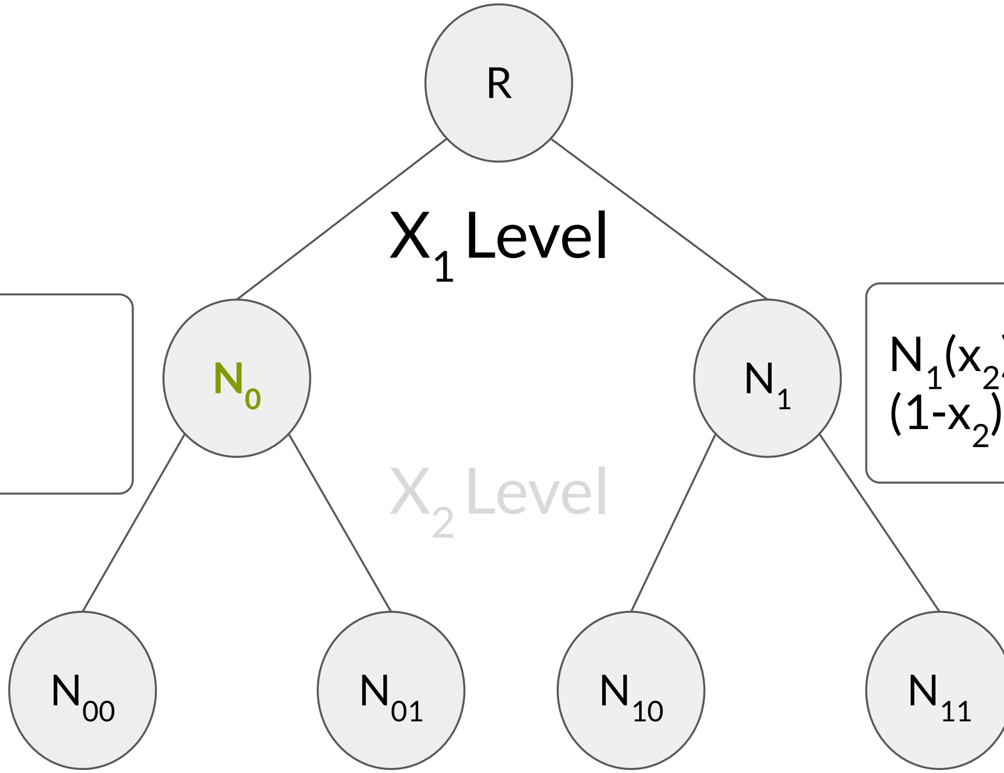
$N_0(x_2) =$
$(1-x_2)N_{00}+x_2N_{01}$



$N_1(x_2) =$
$(1-x_2)N_{10}+x_2N_{11}$

Building an MPHT

$R(x_1,x_2) = (1-x_1)N_0(x_2)+x_1N_1(x_2)$



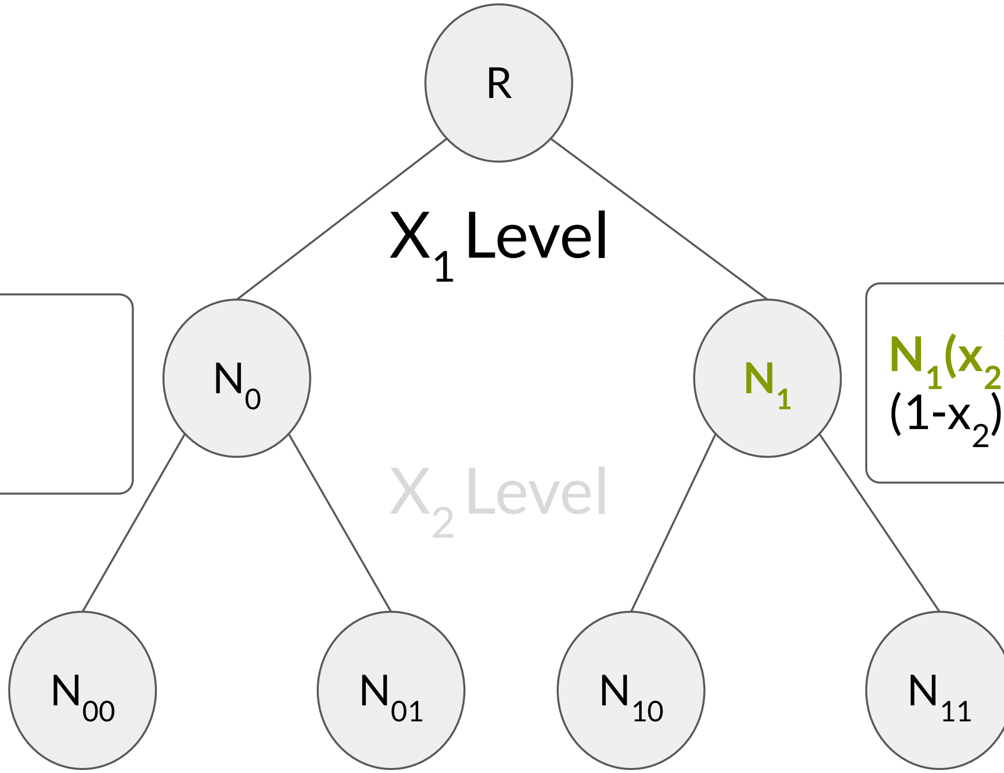
$N_0(x_2) =$
$(1-x_2)N_{00}+x_2N_{01}$

$N_1(x_2) =$
$(1-x_2)N_{10}+x_2N_{11}$

Building an MPHT

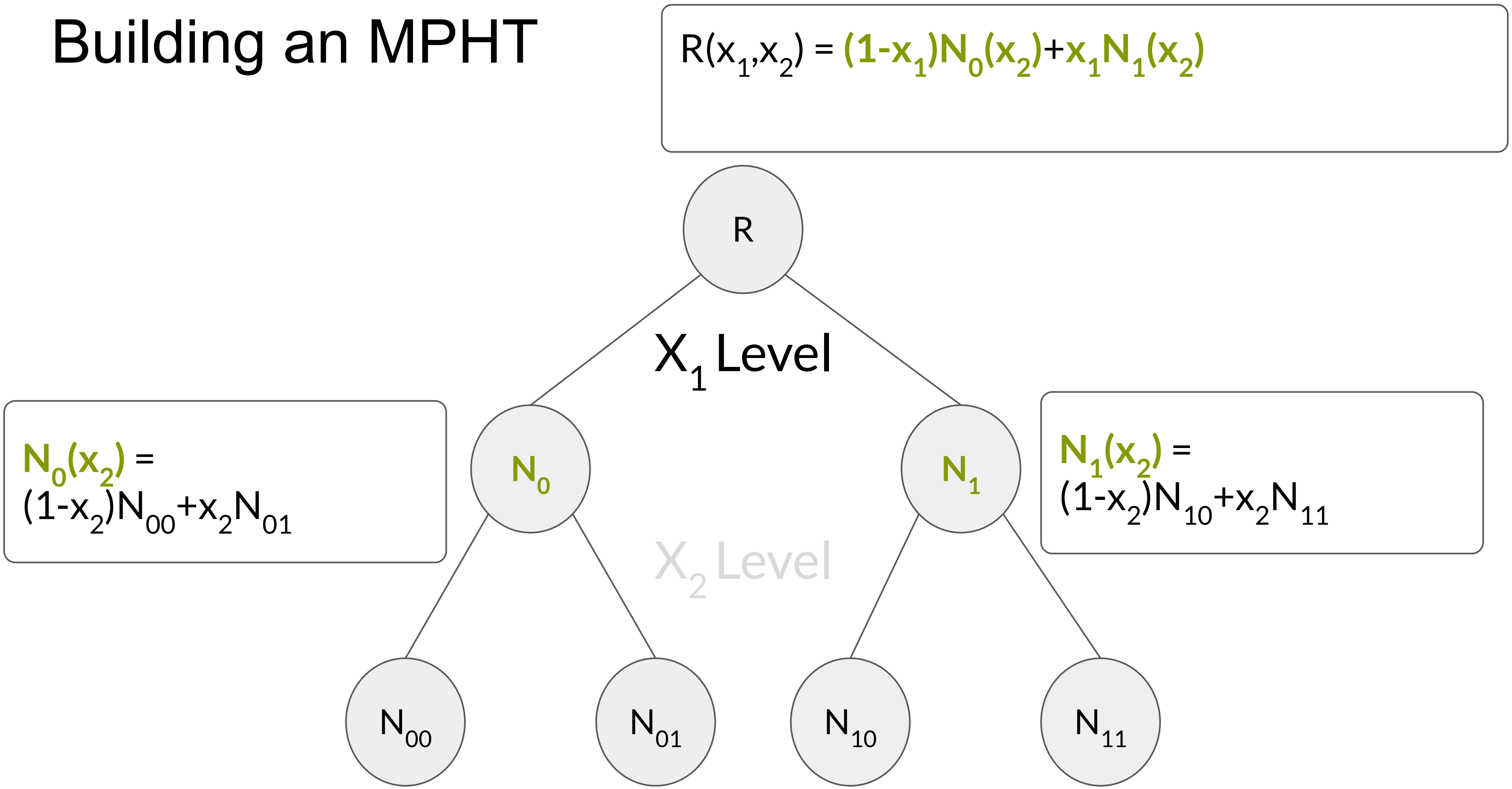
$R(x_1,x_2) = (1-x_1)N_0(x_2)+\mathbf{x_1N_1(x_2)}$

$N_0(x_2) =$
$(1-x_2)N_{00}+x_2N_{01}$



$\mathbf{N_1(x_2)} =$
$(1-x_2)N_{10}+x_2N_{11}$

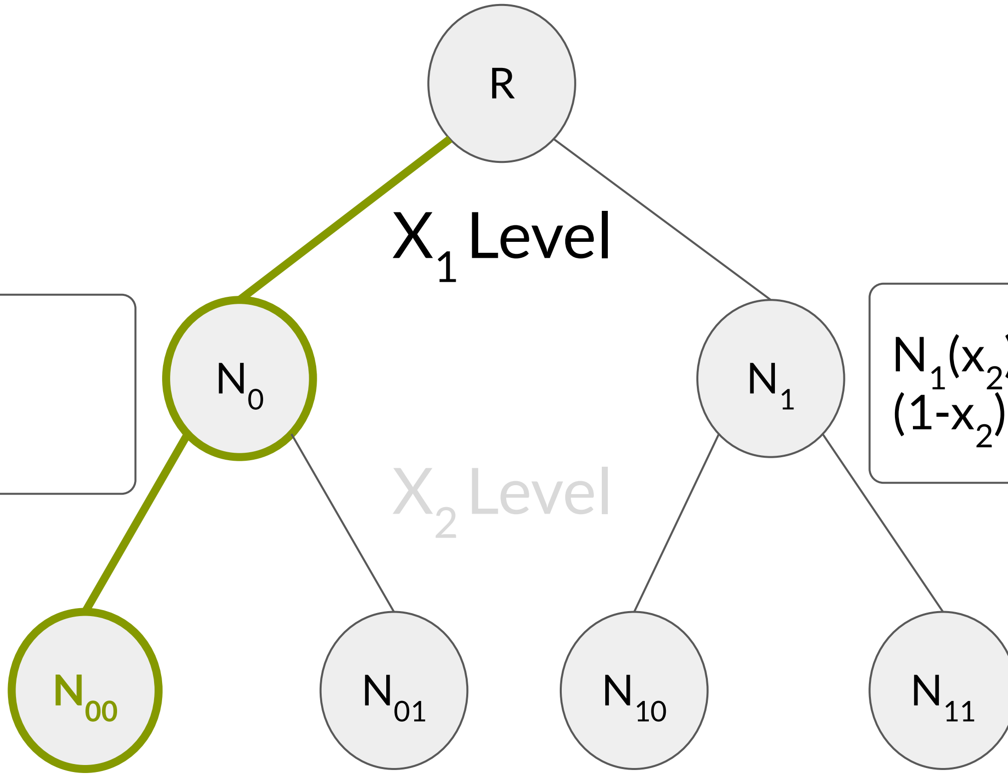
Building an MPHT



Building an MPHT

$R(x_1,x_2) = x_1 N_0(x_2)+(1-x_1)N_1(x_2) =$
$(1-x_1)(1-x_2)N_{00}+(1-x_1)x_2N_{01}+x_1(1-x_2)N_{10}+x_1x_2N_{11}$

$N_0(x_2) =$
$(1-x_2)N_{00}+x_2N_{01}$

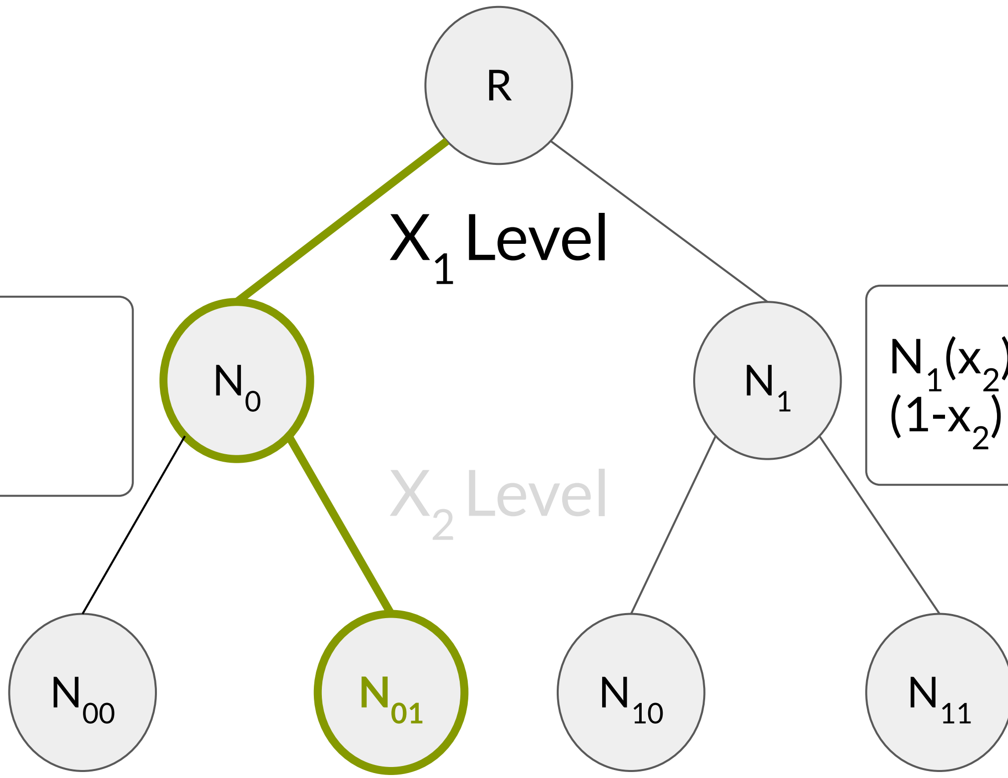


$N_1(x_2) =$
$(1-x_2)N_{10}+x_2N_{11}$

Building an MPHT

$$R(x_1,x_2) = x_1 N_0(x_2)+(1-x_1)N_1(x_2) =$$
$$(1-x_1)(1-x_2)N_{00}+\mathbf{(1-x_1)x_2N_{01}}+x_1(1-x_2)N_{10}+x_1x_2N_{11}$$

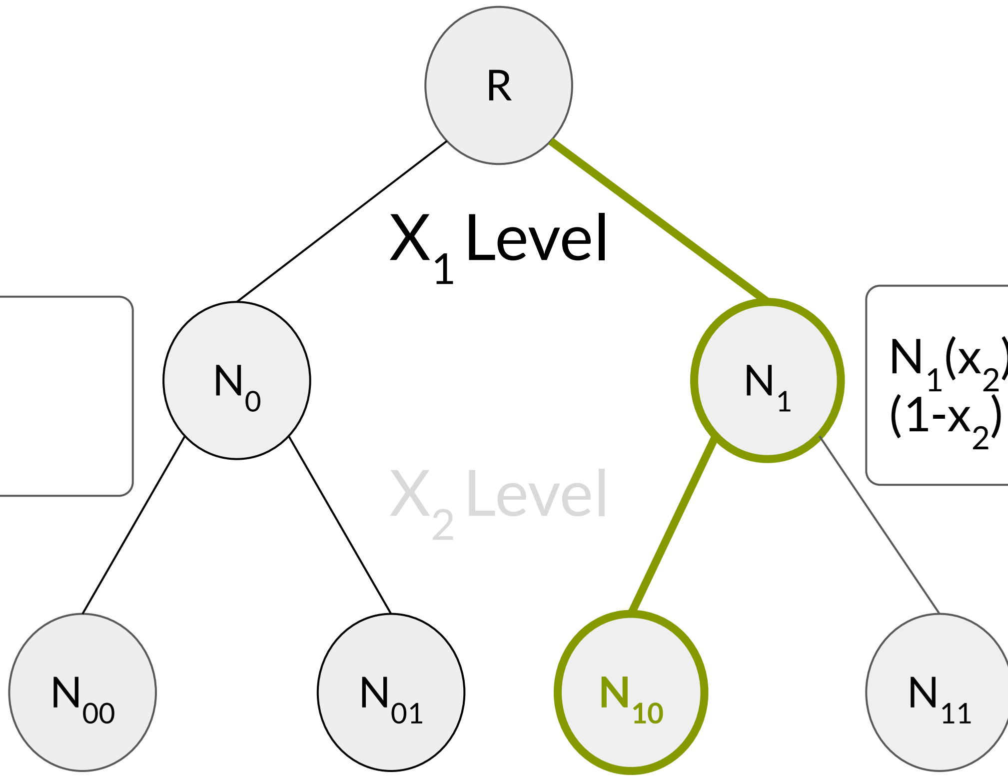
$N_0(x_2) =$
$(1-x_2)N_{00}+x_2N_{01}$



$N_1(x_2) =$
$(1-x_2)N_{10}+x_2N_{11}$

Building an MPHT

$R(x_1,x_2) = x_1N_0(x_2)+(1-x_1)N_1(x_2) =$
$(1-x_1)(1-x_2)N_{00}+(1-x_1)x_2N_{01}+\mathbf{x_1(1-x_2)N_{10}}+x_1x_2N_{11}$



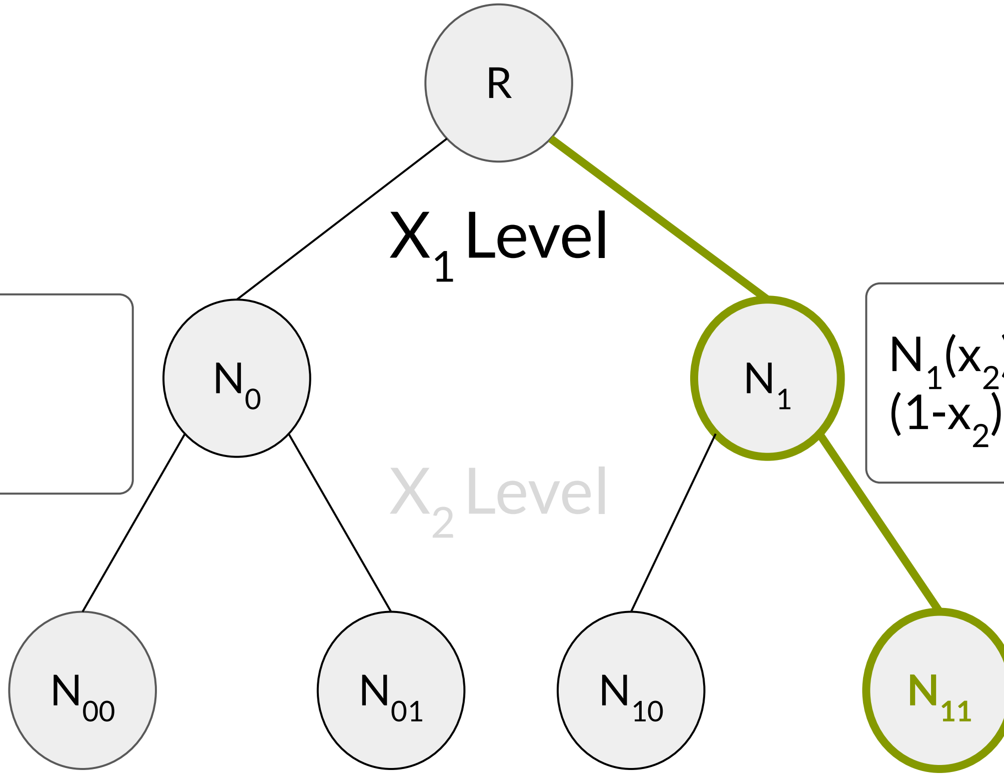
$N_0(x_2) =$
$(1-x_2)N_{00}+x_2N_{01}$

$N_1(x_2) =$
$(1-x_2)N_{10}+x_2N_{11}$

Building an MPHT

$$R(x_1,x_2) = x_1N_0(x_2)+(1-x_1)N_1(x_2) =$$
$$(1-x_1)(1-x_2)N_{00}+(1-x_1)x_2N_{01}+x_1(1-x_2)N_{10}+x_1x_2N_{11}$$

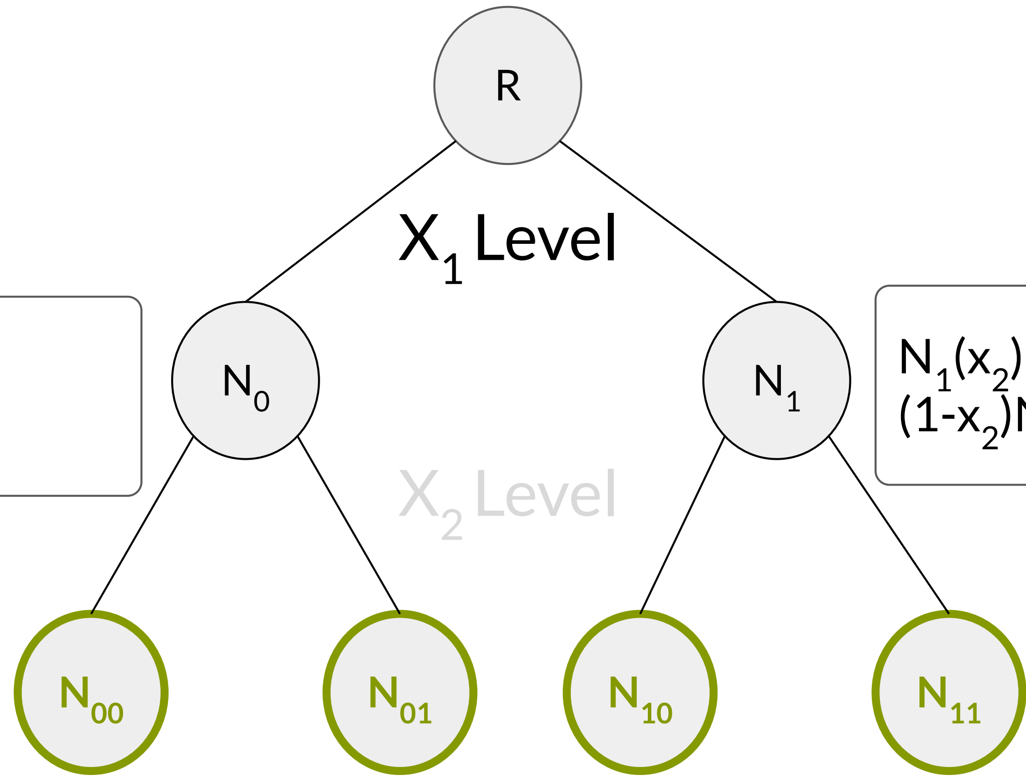
$N_0(x_2) =$
$(1-x_2)N_{00}+x_2N_{01}$



$N_1(x_2) =$
$(1-x_2)N_{10}+x_2N_{11}$

Building an MPHT

$R(x_1,x_2) = x_1N_0(x_2)+(1-x_1)N_1(x_2) =$
$(1-x_1)(1-x_2)N_{00}+(1-x_1)x_2N_{01}+x_1(1-x_2)N_{10}+x_1x_2N_{11}$

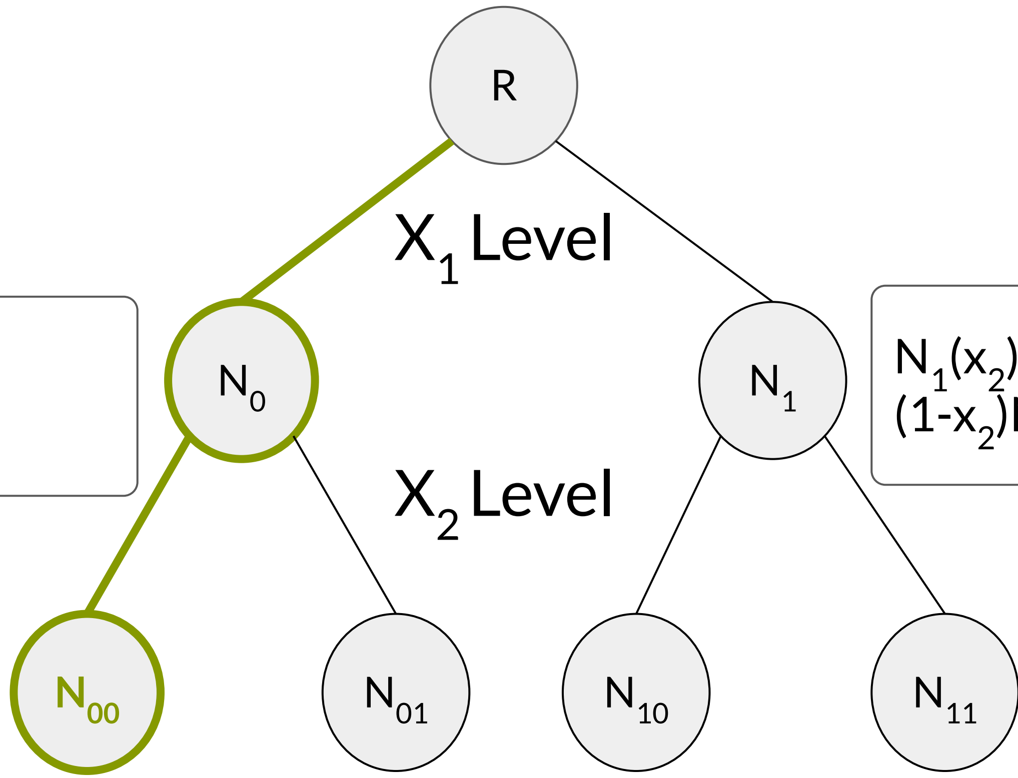


$N_0(x_2) =$
$(1-x_2)N_{00}+x_2N_{01}$

$N_1(x_2) =$
$(1-x_2)N_{10}+x_2N_{11}$

Building an MPHT

$R(x_1,x_2) = x_1N_0(x_2)+(1-x_1)N_1(x_2) =$
$(1-x_1)(1-x_2)N_{00}+(1-x_1)x_2N_{01}+x_1(1-x_2)N_{10}+x_1x_2N_{11}$



$N_0(x_2) =$
$(1-x_2)N_{00}+x_2N_{01}$

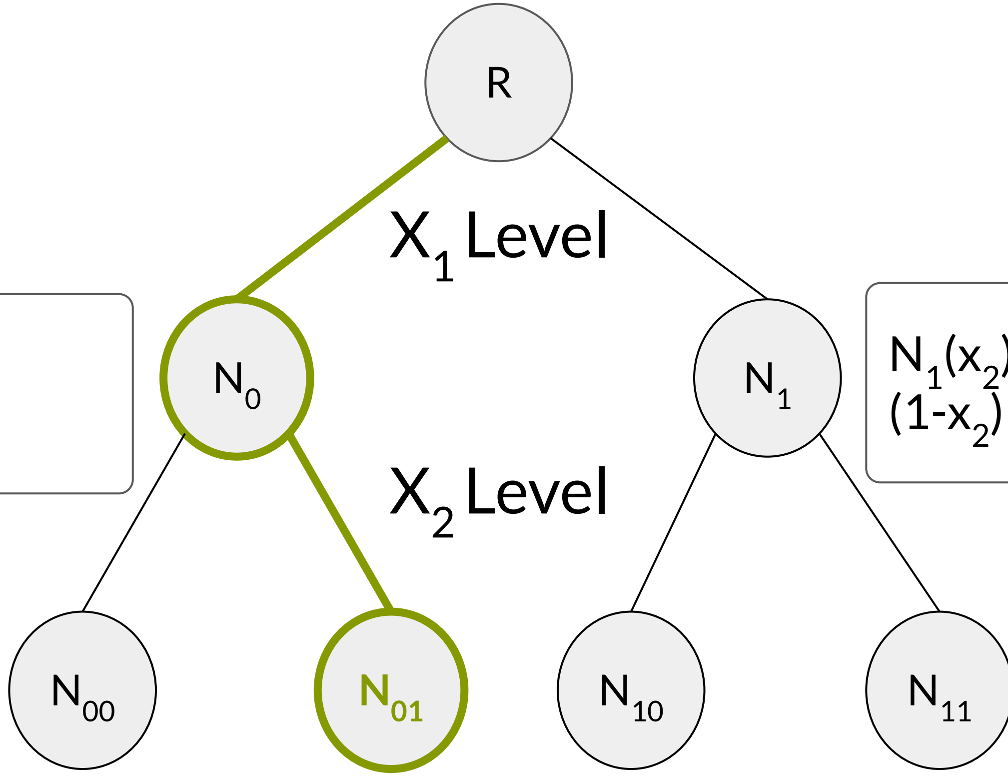
$N_1(x_2) =$
$(1-x_2)N_{10}+x_2N_{11}$

$R(0,0)=N_{00}$

Building an MPHT

$$R(x_1,x_2) = x_1N_0(x_2)+(1-x_1)N_1(x_2) = (1-x_1)(1-x_2)N_{00}+(1-x_1)x_2N_{01}+x_1(1-x_2)N_{10}+x_1x_2N_{11}$$

$$N_0(x_2) = (1-x_2)N_{00}+x_2N_{01}$$

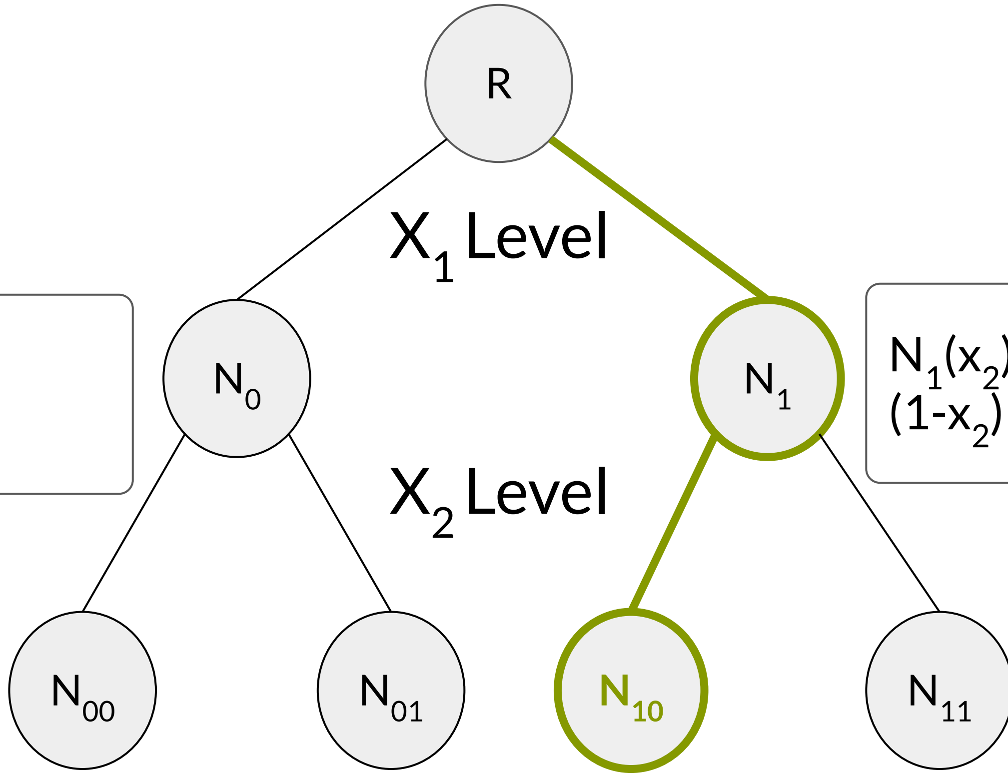


$$N_1(x_2) = (1-x_2)N_{10}+x_2N_{11}$$

$$R(0,1)=N_{01}$$

Building an MPHT

$$R(x_1,x_2) = x_1 N_0(x_2)+(1-x_1)N_1(x_2) =$$
$$(1-x_1)(1-x_2)N_{00}+(1-x_1)x_2N_{01}+\mathbf{x_1(1-x_2)N_{10}}+x_1x_2N_{11}$$



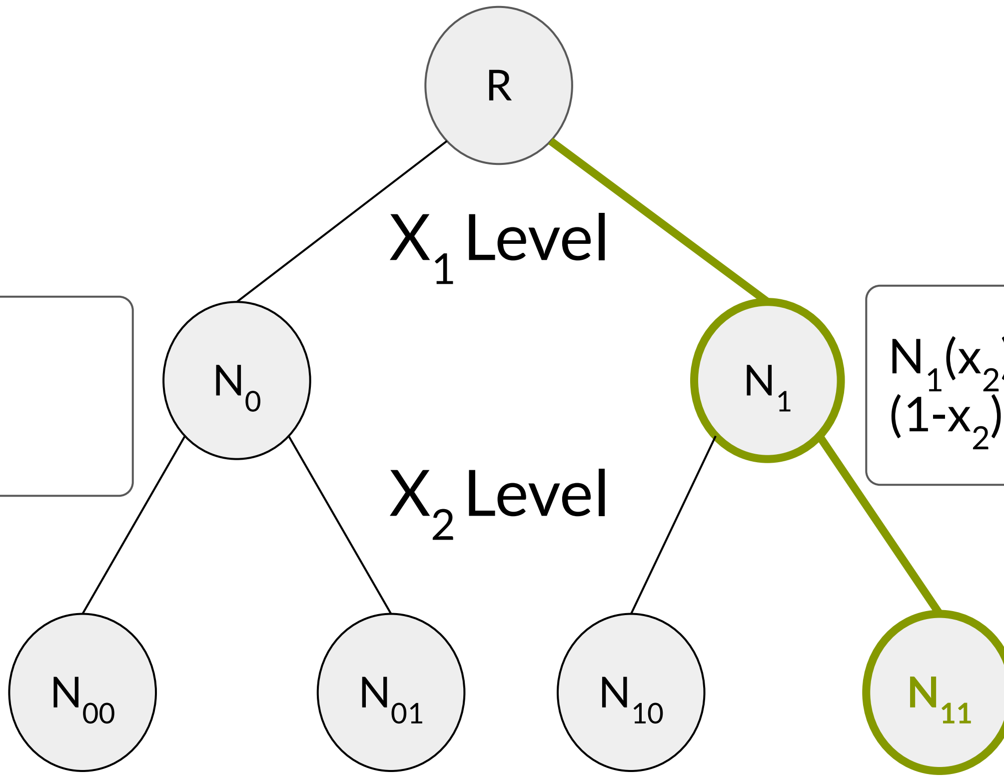
$$N_0(x_2) =$$
$$(1-x_2)N_{00}+x_2N_{01}$$

$$N_1(x_2) =$$
$$(1-x_2)N_{10}+x_2N_{11}$$

$$R(1,0)=\mathbf{N_{10}}$$

Building an MPHT

$$R(x_1,x_2) = x_1 N_0(x_2)+(1-x_1)N_1(x_2) =$$
$$(1-x_1)(1-x_2)N_{00}+(1-x_1)x_2N_{01}+x_1(1-x_2)N_{10}+x_1x_2N_{11}$$

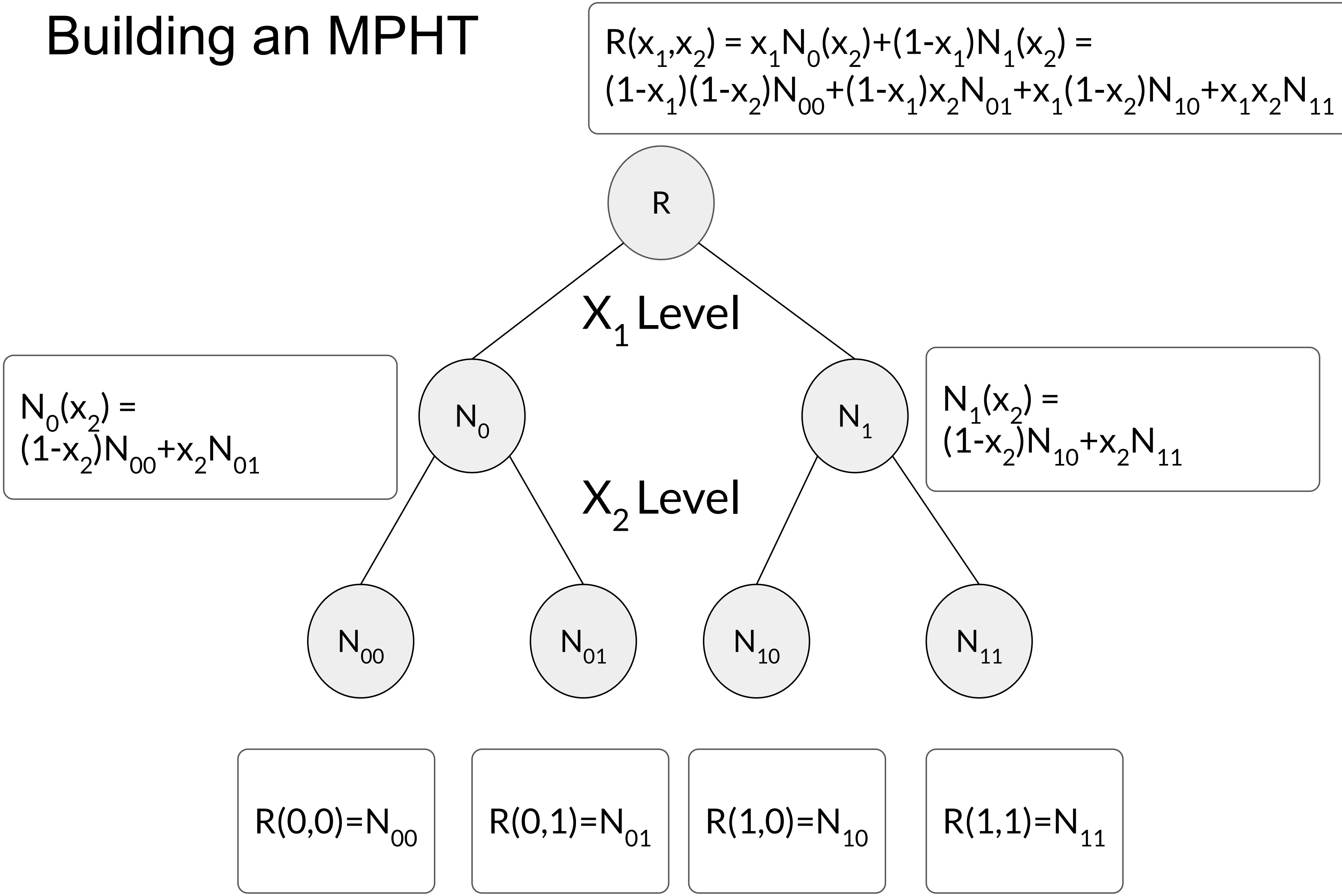


$$N_0(x_2) = (1-x_2)N_{00}+x_2N_{01}$$

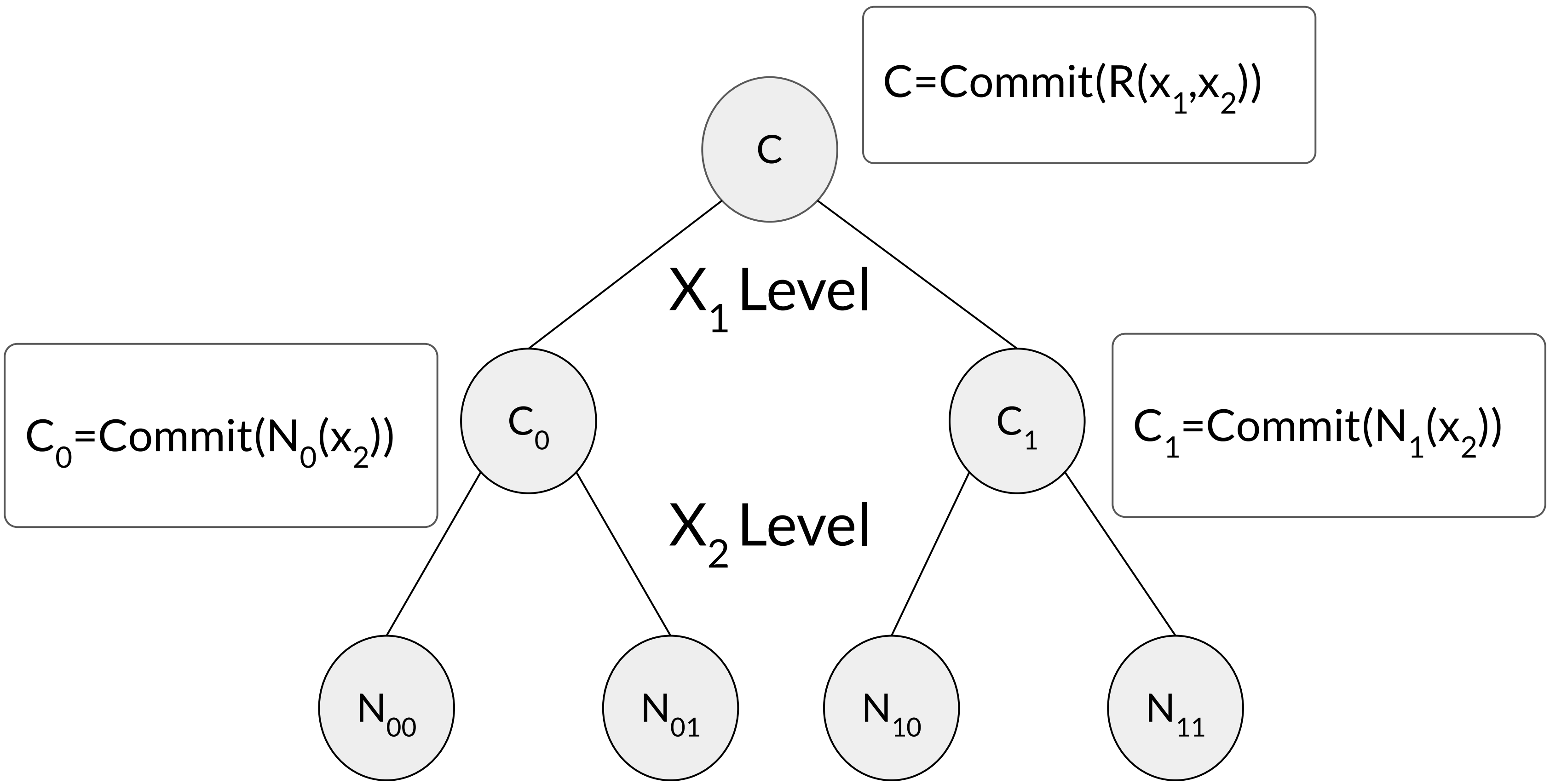
$$N_1(x_2) = (1-x_2)N_{10}+x_2N_{11}$$

$$R(1,1)=N_{11}$$

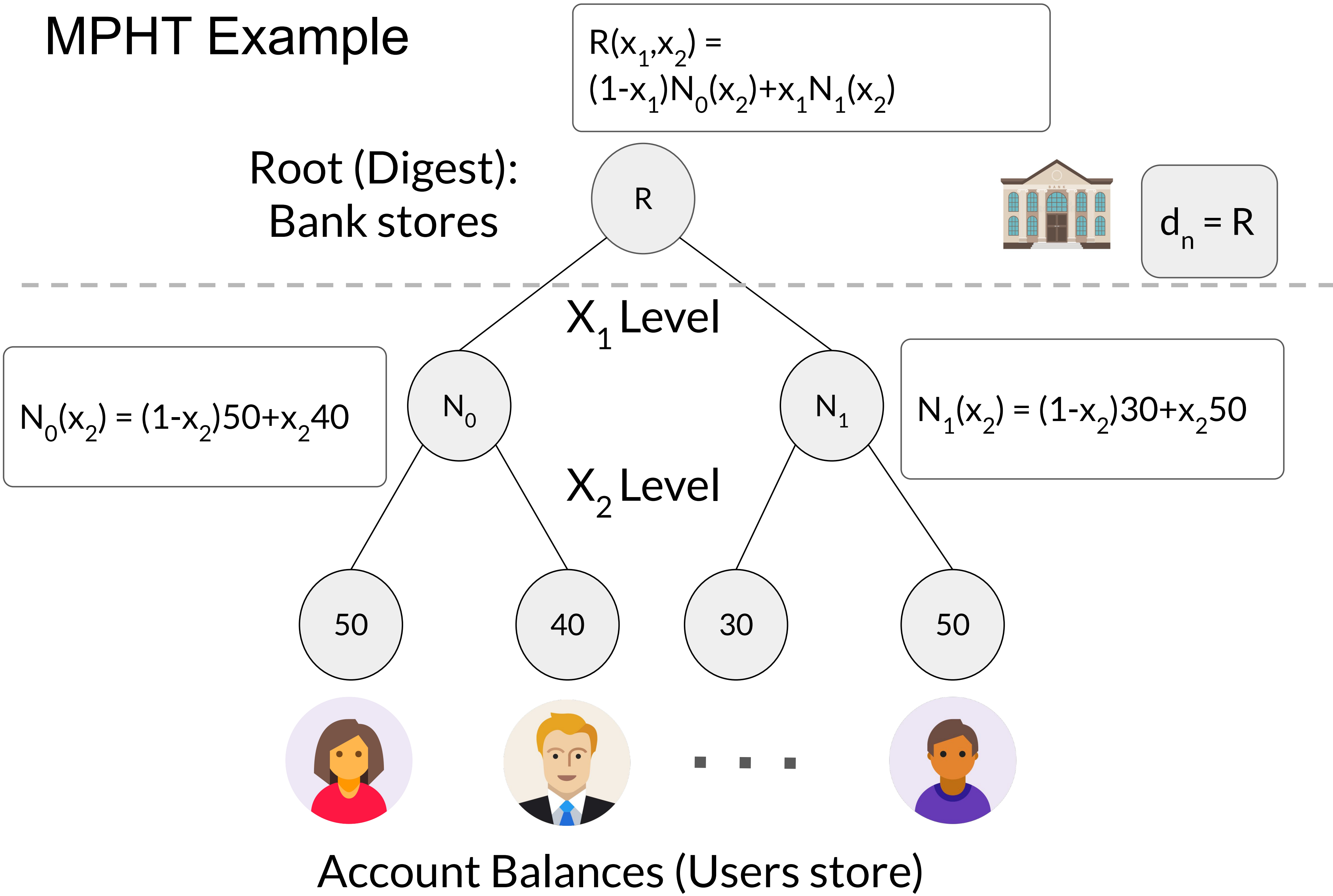
Building an MPHT



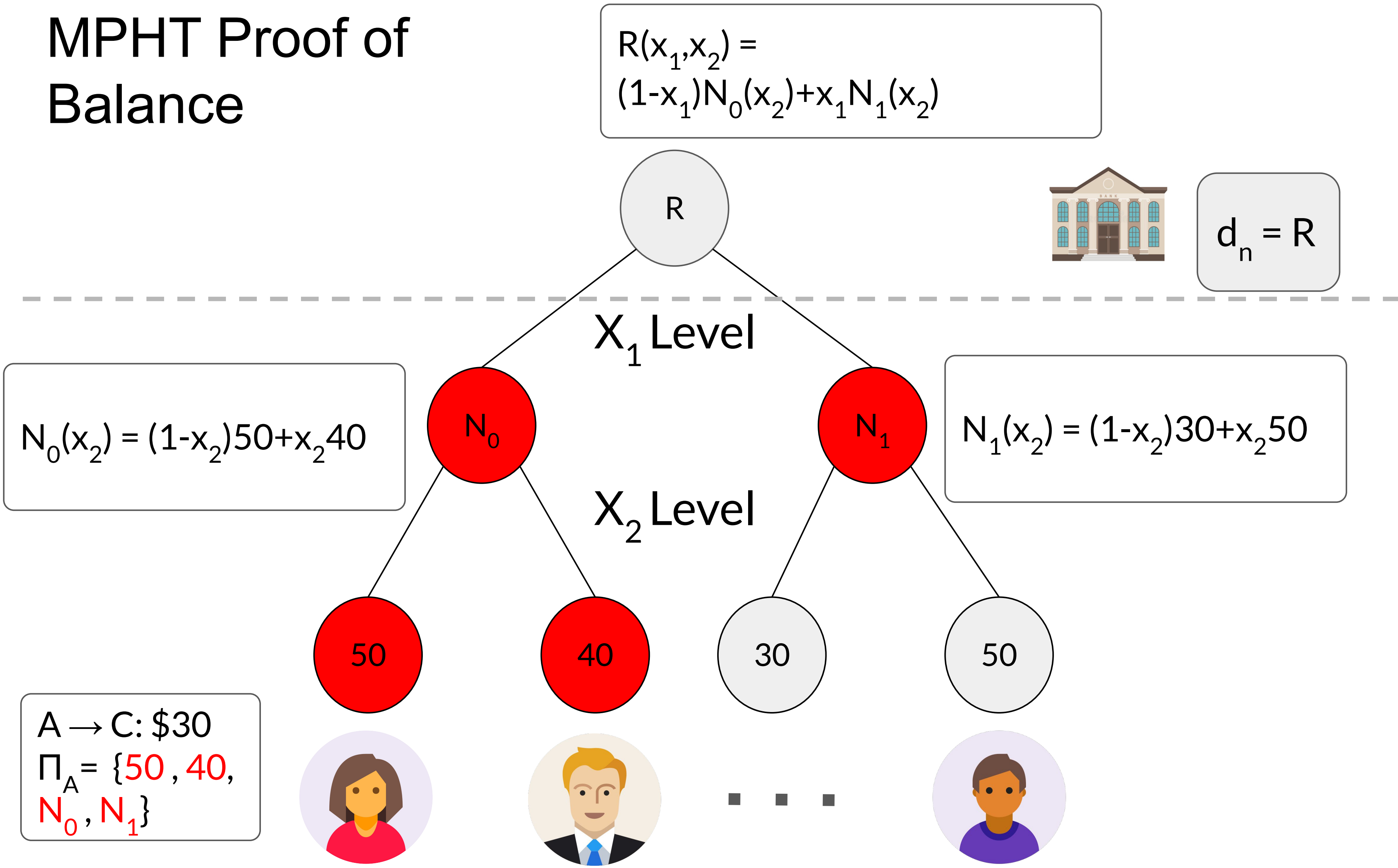
MPHT Commitments to Polynomials



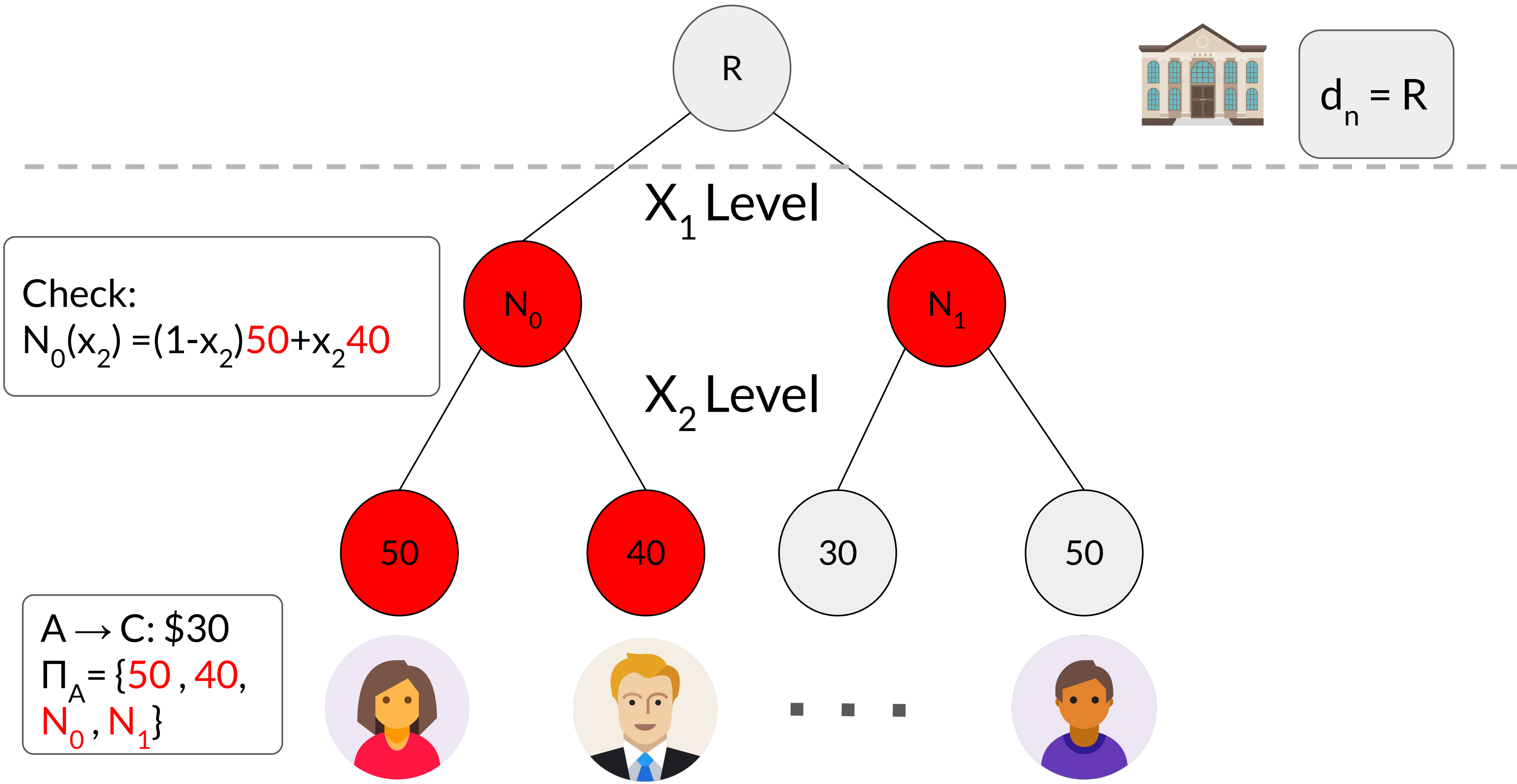
MPHT Example



MPHT Proof of Balance

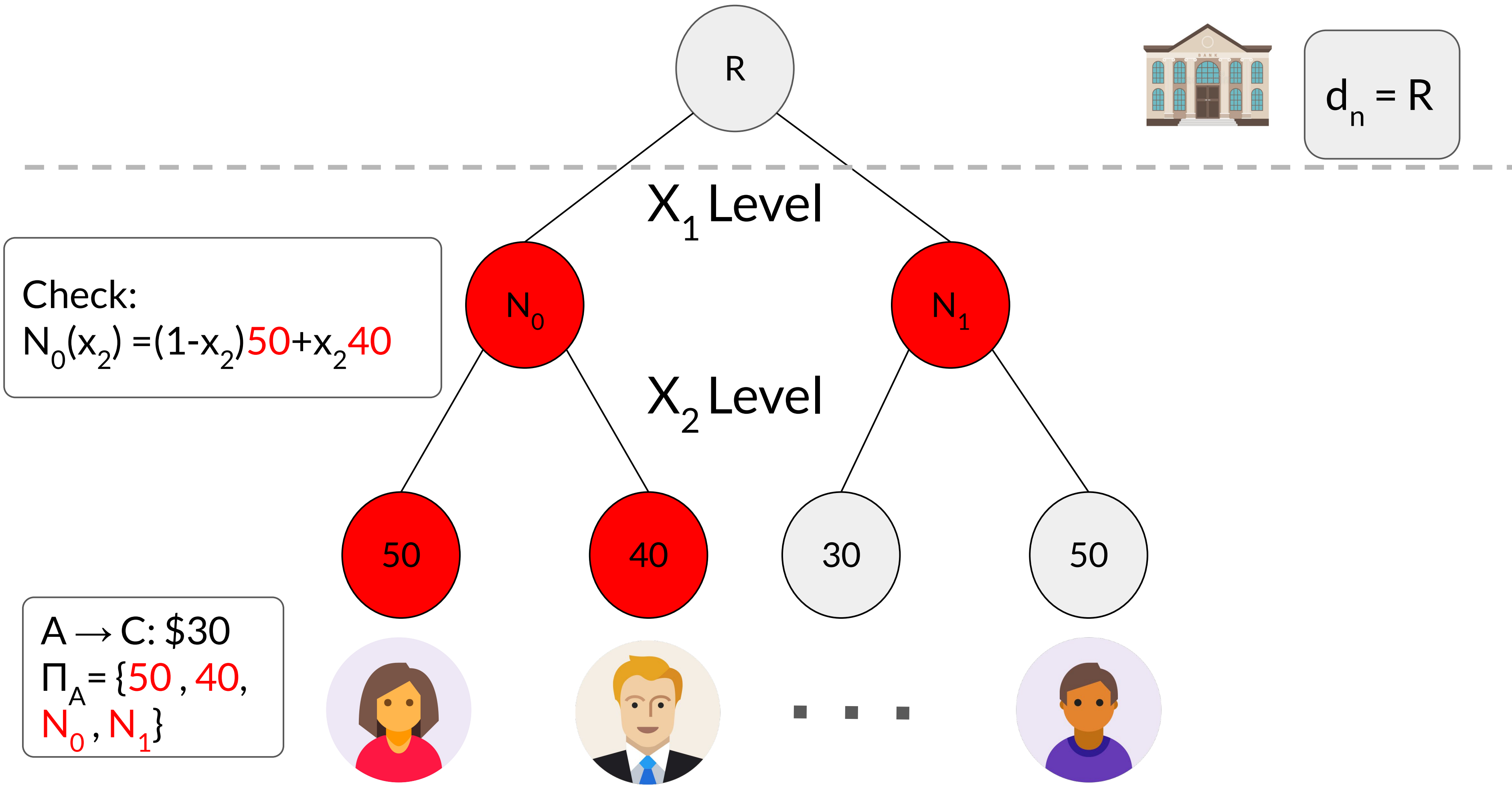


MPHT Proof of Balance Verification

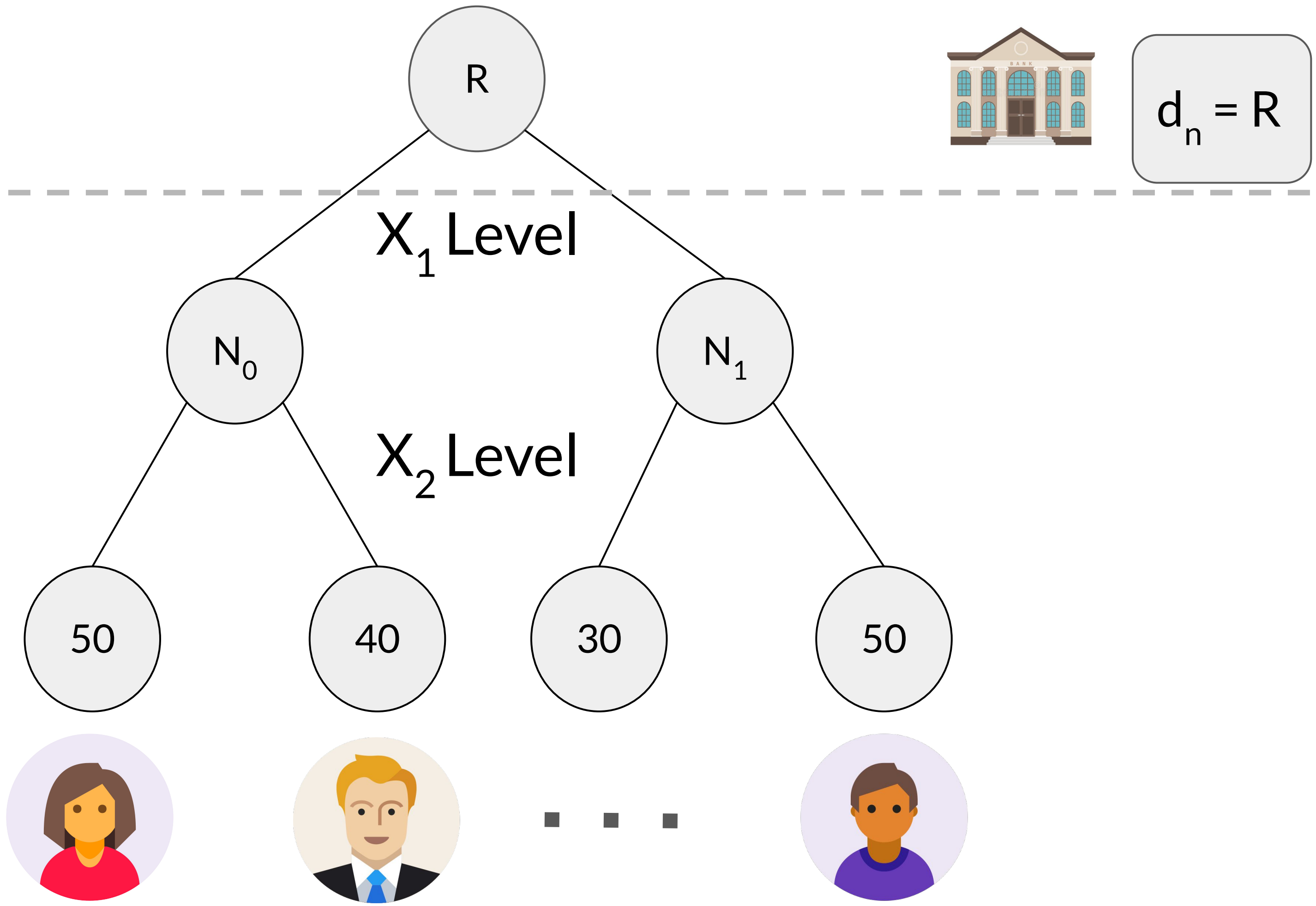


MPHT Proof of Balance Verification

Check:
$R(x_1,x_2) = (1-x_1)N_0(x_2)+x_1N_1(x_2)$

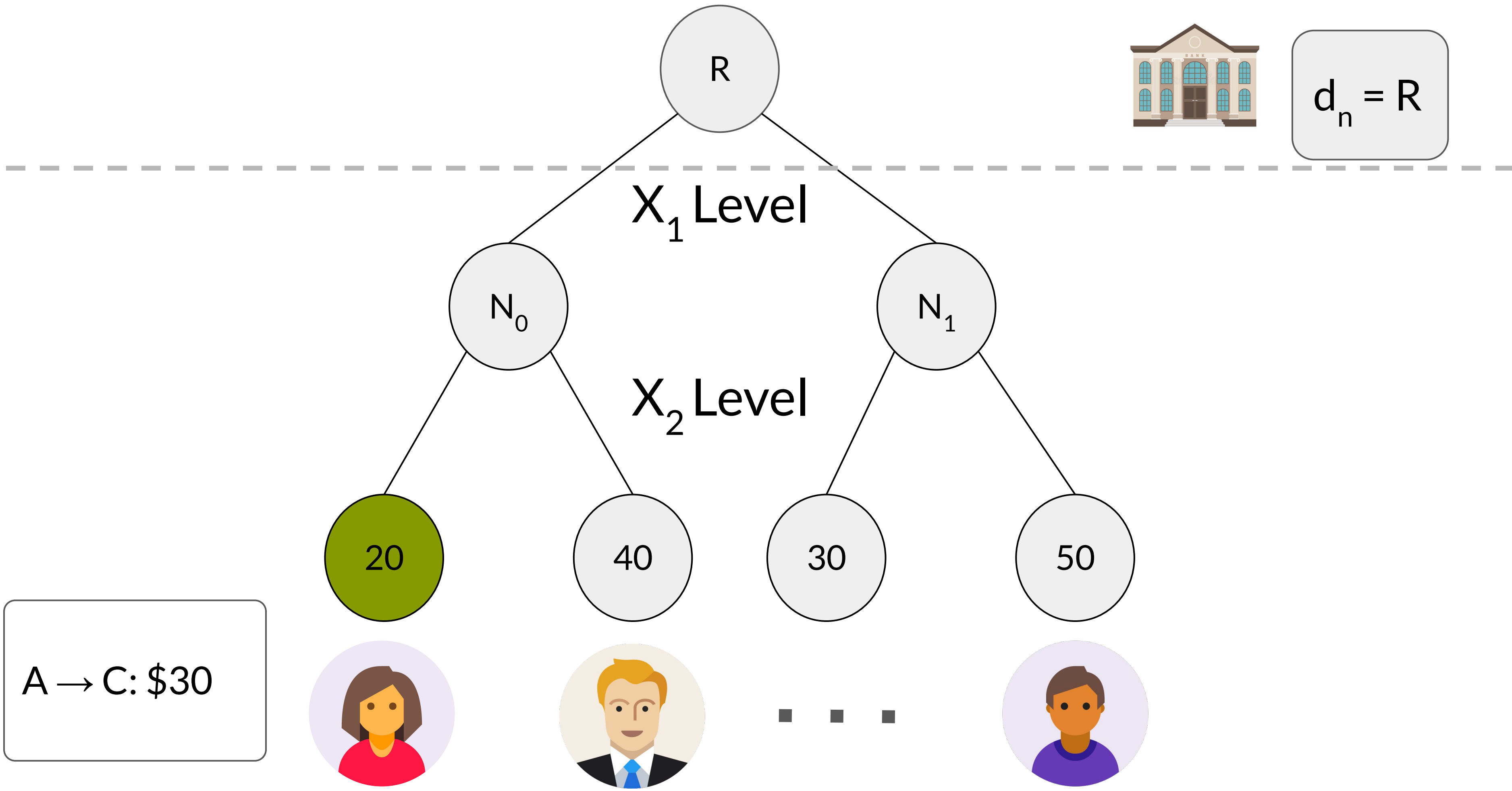


MPHT Updating Digest



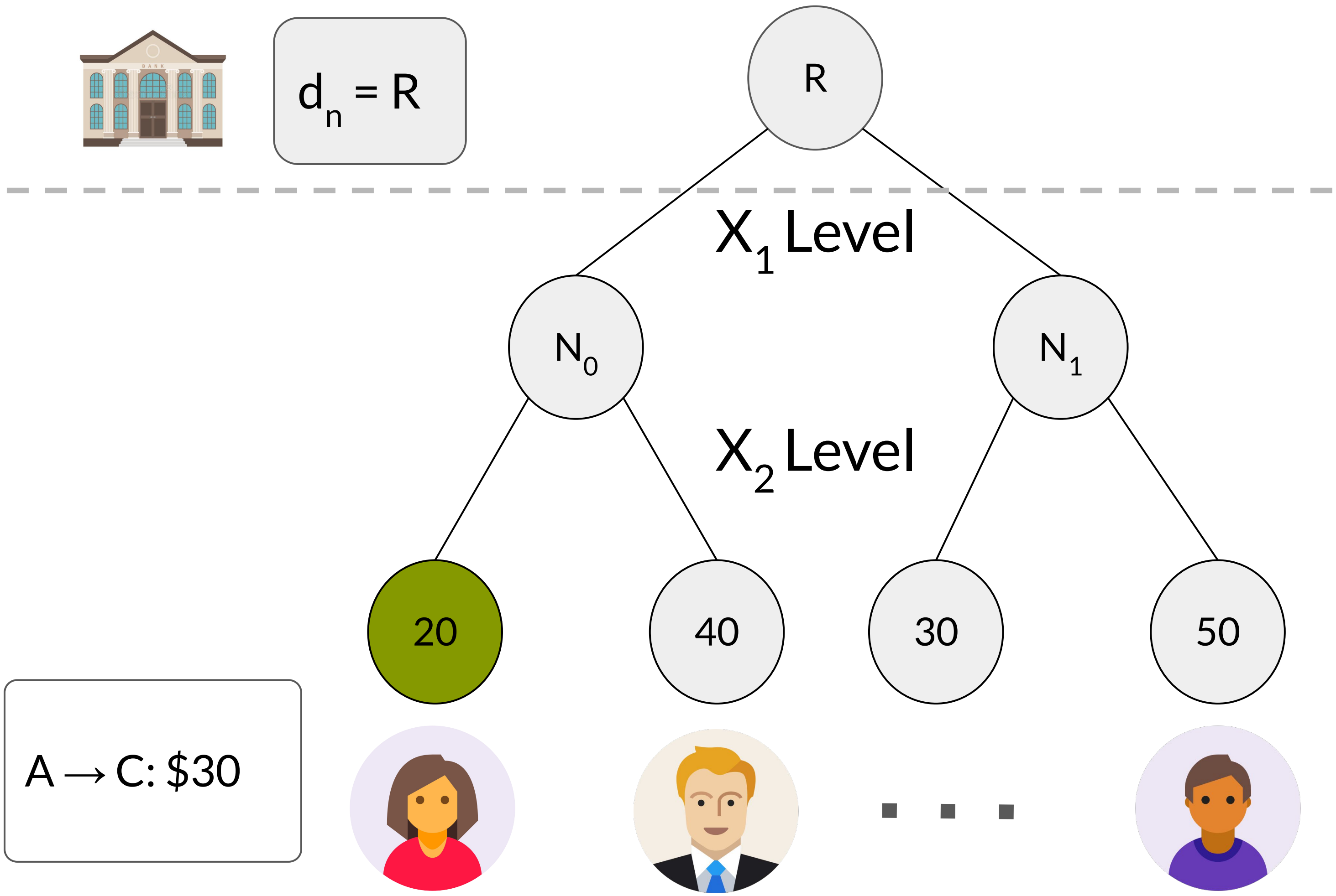
A → C: $30

MPHT Updating Digest

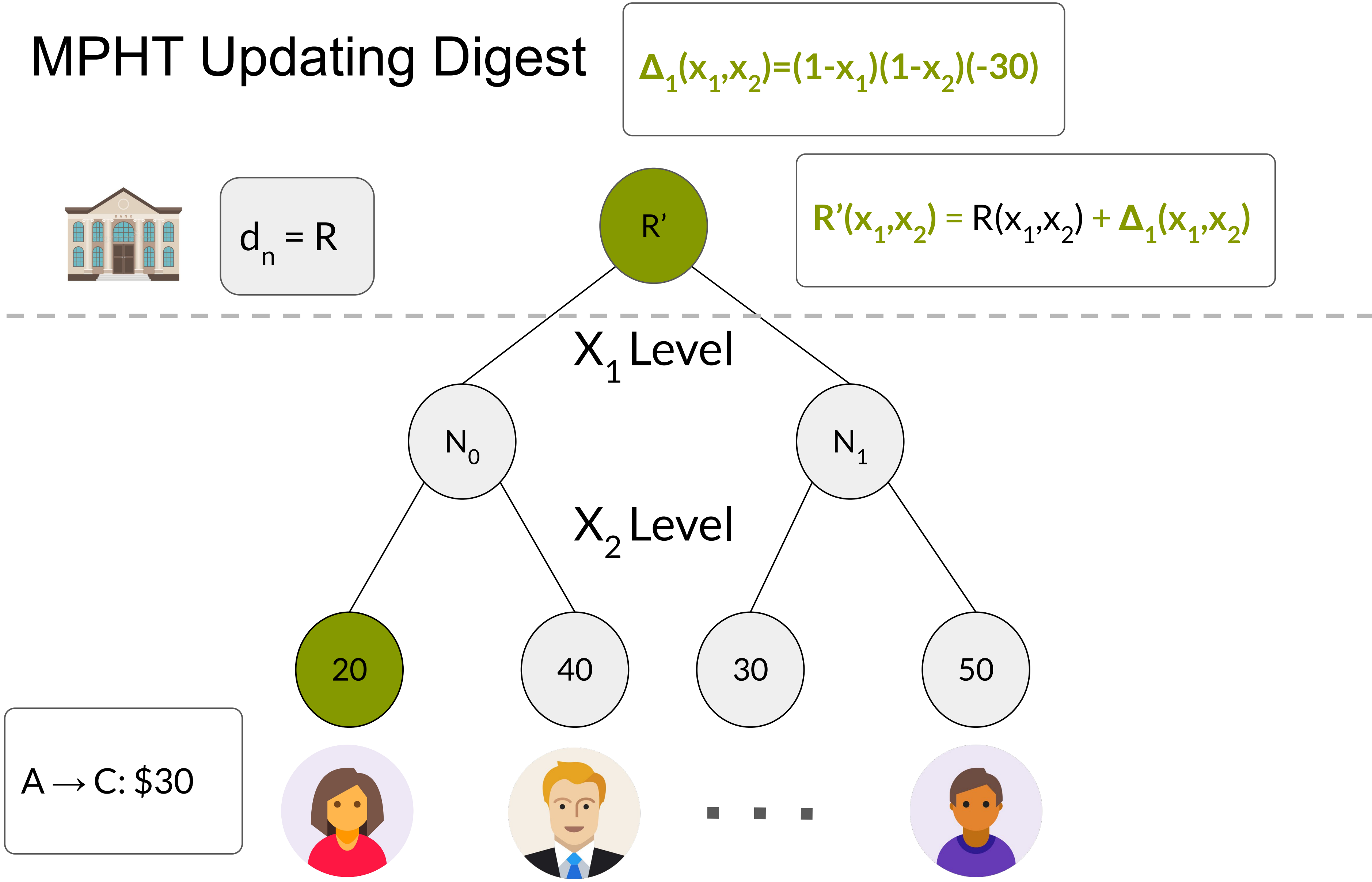


MPHT Updating Digest

$$\Delta_1(x_1,x_2)=(1-x_1)(1-x_2)(-30)$$

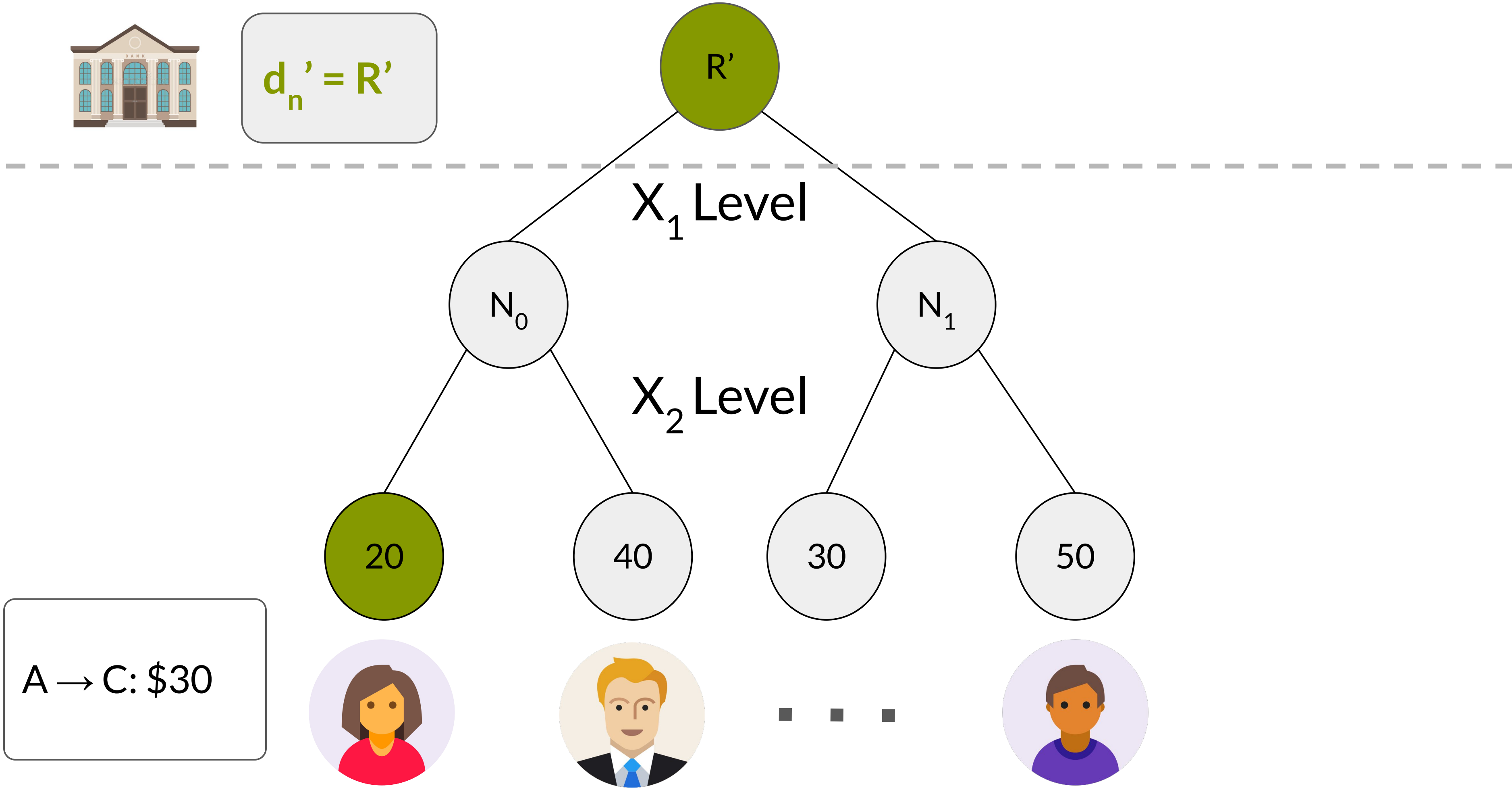


MPHT Updating Digest



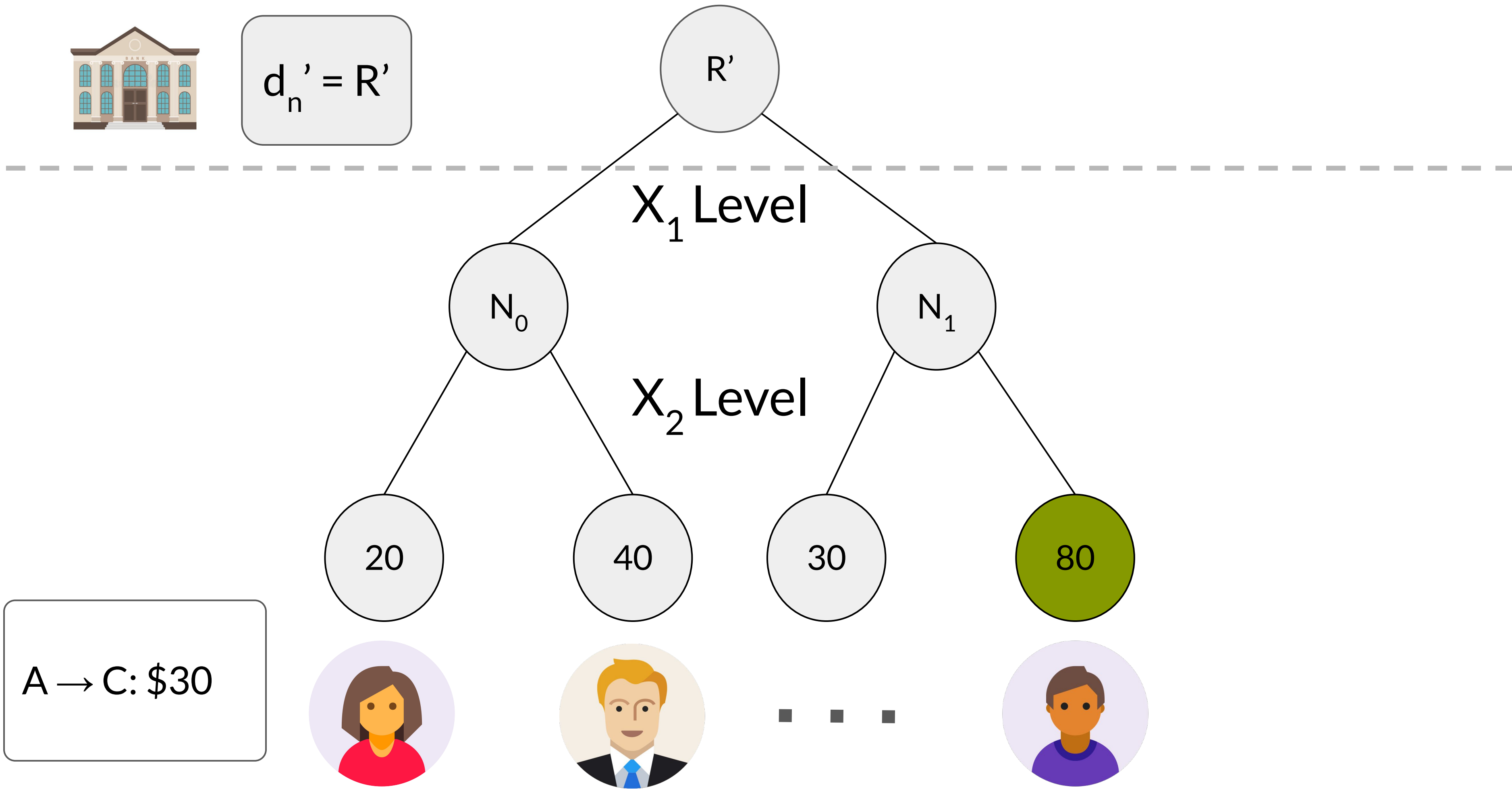
MPHT Updating Digest

$$R'(x_1,x_2)$$
$$=(1-x_1)(1-x_2)50+(1-x_1)x_2 40+x_1(1-x_2)30+x_1 x_2 50+$$
$$(1-x_1)(1-x_2)(-30)$$
$$=(1-x_1)(1-x_2)20+(1-x_1)x_2 40+x_1(1-x_2)30+x_1 x_2 50$$



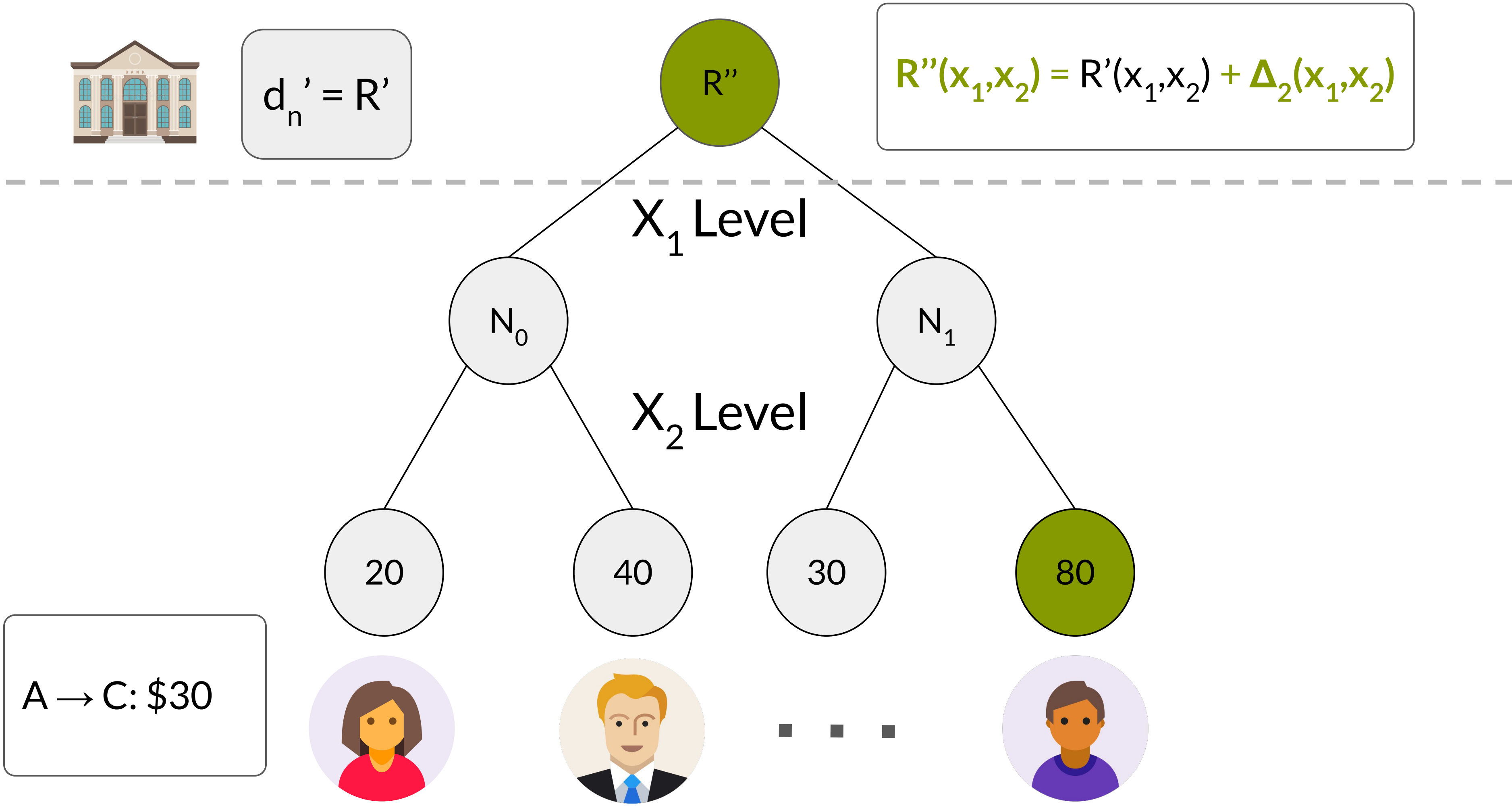
MPHT Updating Digest

$\Delta_2(x_1,x_2)=+x_1x_2 30$



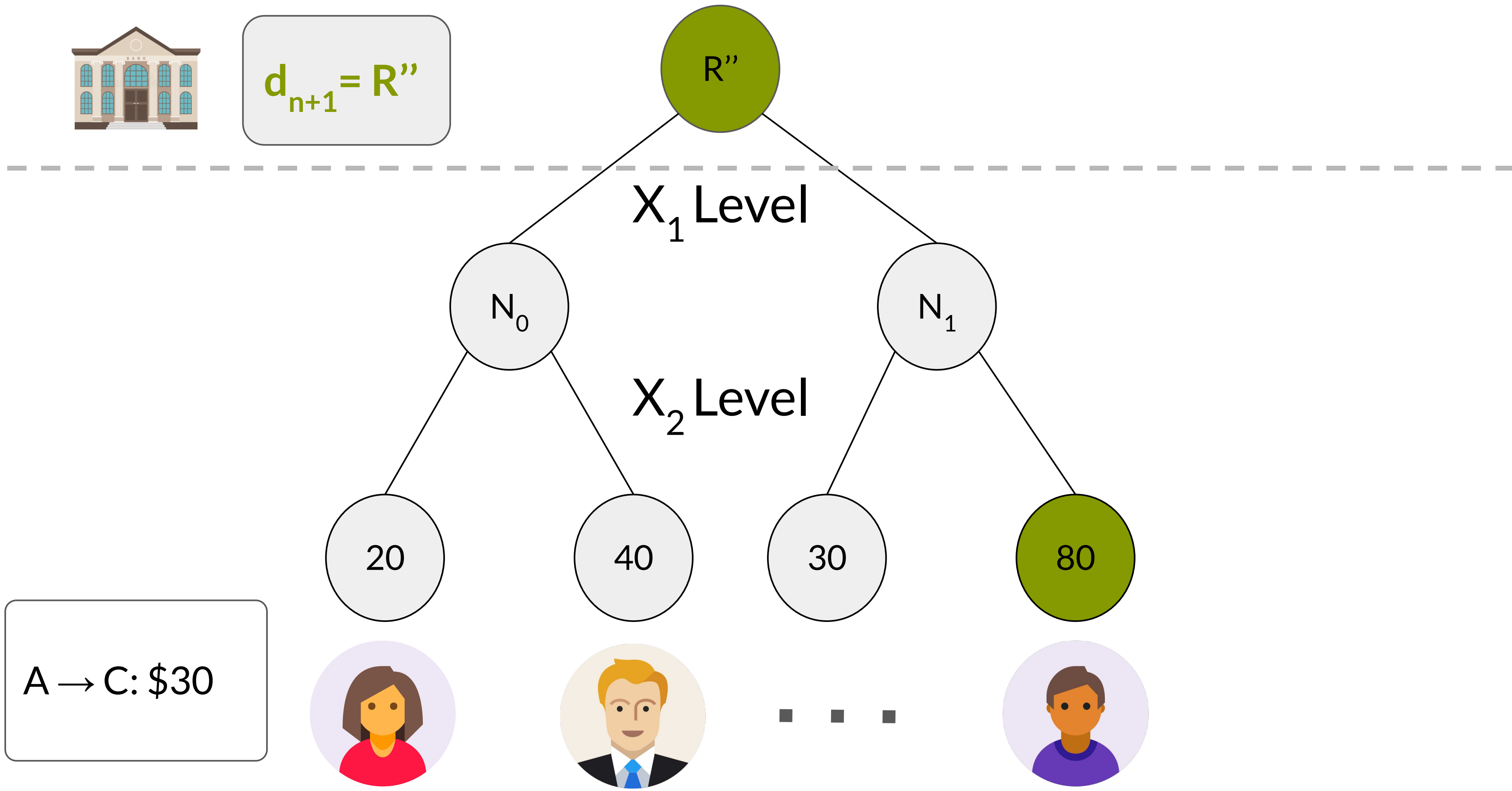
MPHT Updating Digest

$\Delta_2(x_1,x_2) = +x_1x_2 30$



MPHT Updating Digest

$$R'(x_1,x_2)$$
$$=(1-x_1)(1-x_2)20+(1-x_1)x_2 40+x_1(1-x_2)30+x_1x_2 50$$
$$+x_1x_2 30$$
$$=(1-x_1)(1-x_2)20+(1-x_1)x_2 40+x_1(1-x_2)30+x_1x_2 80$$



MPHT Updating Proofs

- Out of time
- High-level idea:
  - There exist *"public parameters"*
  - Clients use them to update their proofs of balance after seeing transactions

Conclusion

- We present a new type of Merkle tree based on multivariate polynomials with an efficiently updatable digest
- Can be used to scale TXN verifications in cryptocurrencies (e.g. Ethereum)

Drawbacks/Future Work

- A large number of public parameters are needed in this construction to "hash" multivariate polynomials (however, clients do not need to store them if a fully-untrusted server does)
- Verifying proofs of balance in our tree is more expensive than the MHT construction (~1000x), but should still be much faster than going to disk

Acknowledgements

Thanks to my mentor Alin Tomescu for his support and guidance!

Thanks to PRIMES for this opportunity!

Thanks to my parents for their support!

Thank you!

Questions?