Concurrency as an Iterated Affine Task

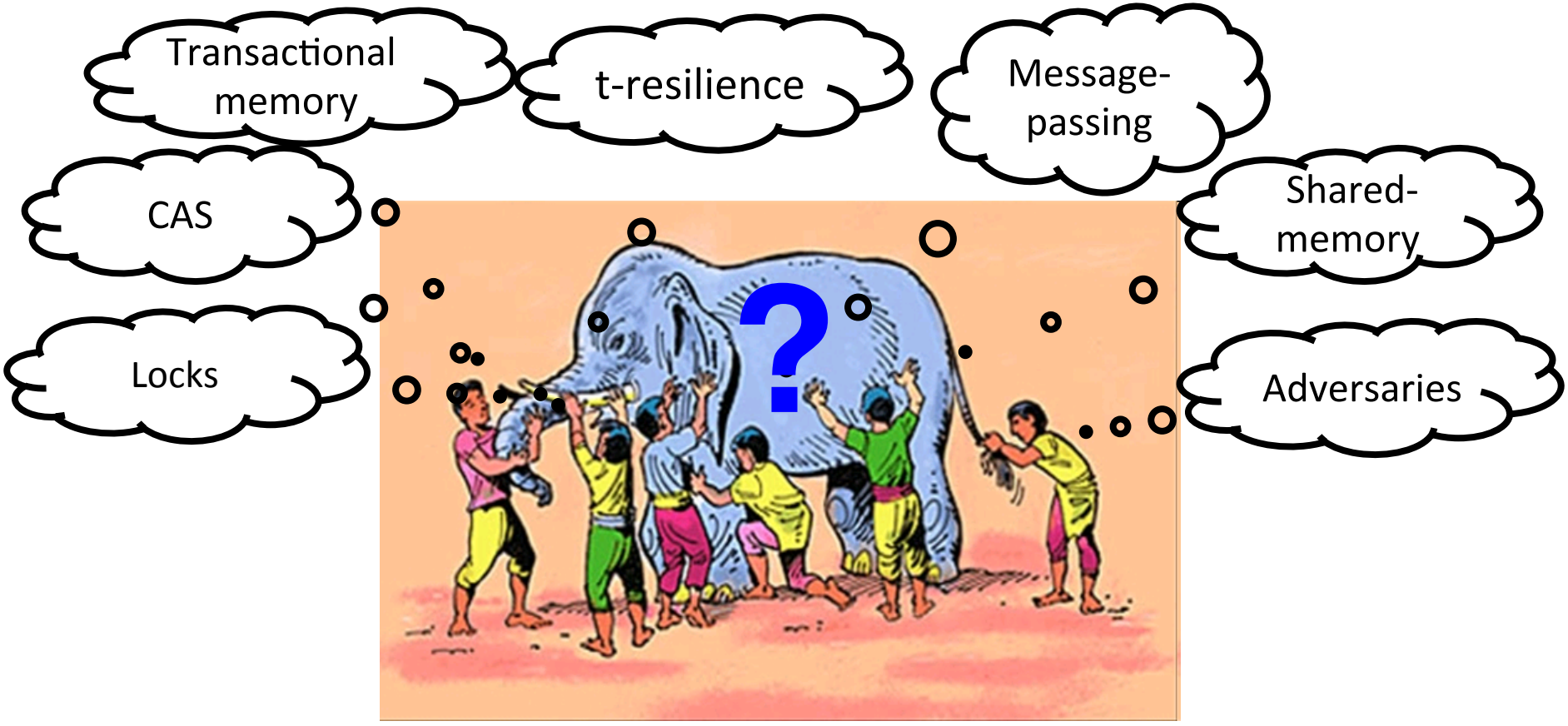
Eli Gafni Yuan He

UCLA

Petr Kuznetsov Thibault Rieutord

Telecom ParisTech

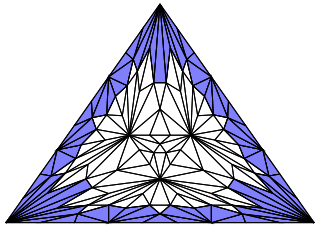
CMO/BIRS workshop, 2016


Transactional memory
t-resilience
Message-passing
CAS
Shared-memory
Locks
Adversaries
?

Takeaways

- A (long-lived, non-compact) model can be matched by a (one-shot, compact) task

- k-concurrency has a matching task
  - ✓ also holds for adversaries
  - ✓“natural” models



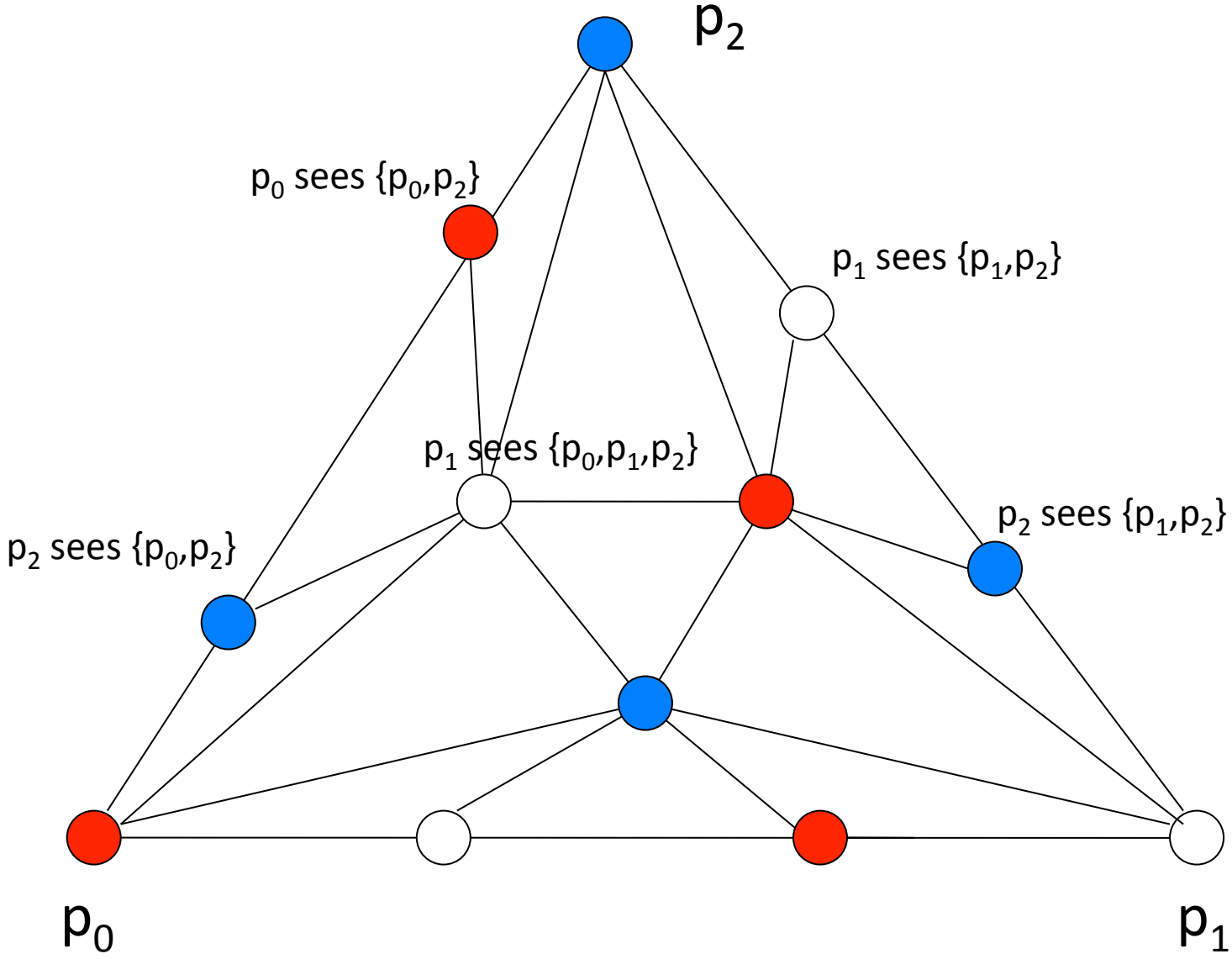
- BG simulation is great!
  - ✓Borowsky&Gafni, 1993
  - ✓….

Example: one-shot Immediate Snapshot

Processes are partitioned in batches $P_1,\ldots,P_m$

- Each batch P_i:
  - ✓Write their input
  - ✓Get a snapshot of the memory
- Gives the standard chromatic subdivision $\chi(s^n)$
  - ✓[BG93]
  - ✓[Koz14,Lin09]

$X(s^2)$: one-shot IS for 3 processes



Iterated Immediate Snapshot (IIS)

Shared variables:

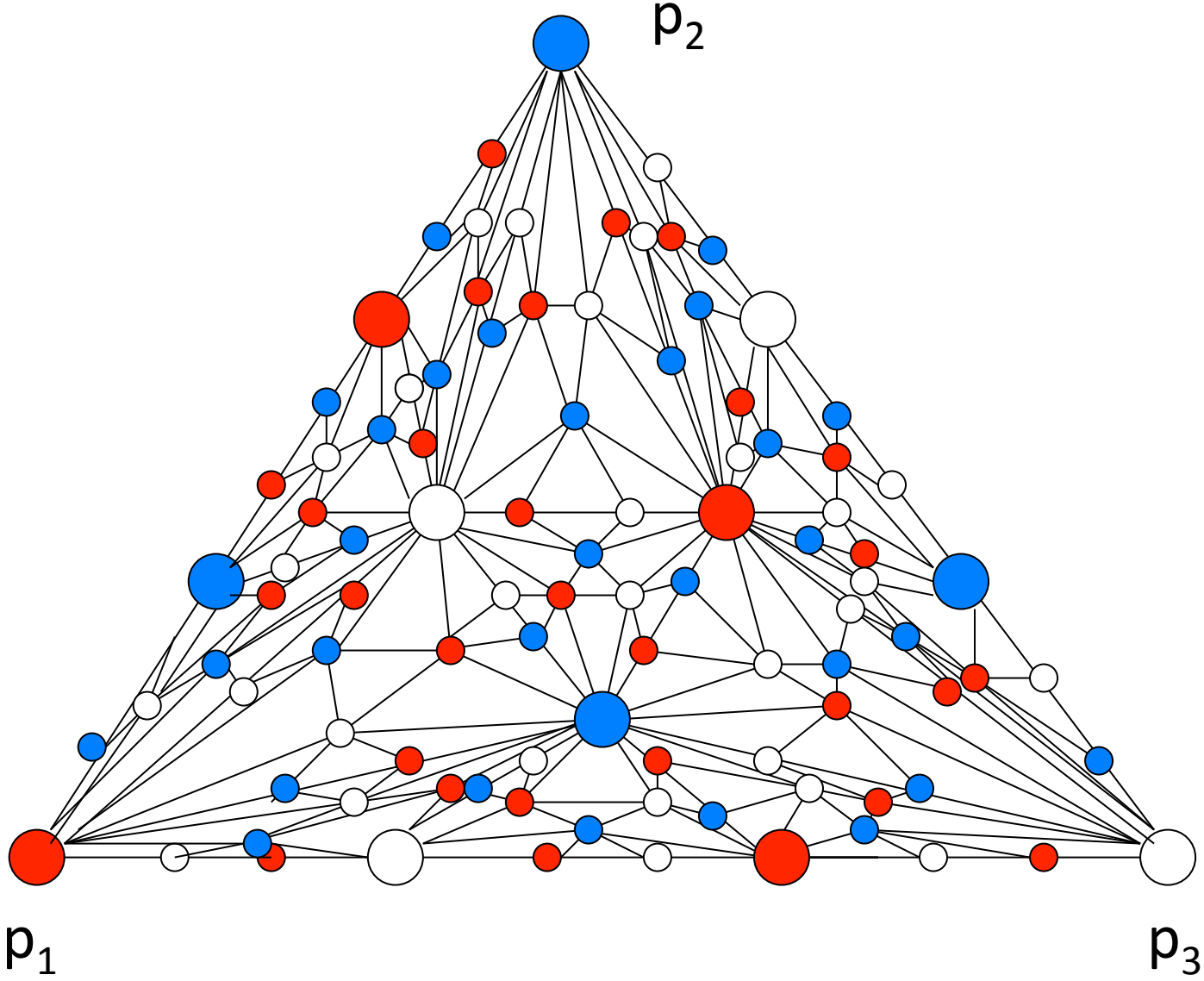
IS_1, IS_2, IS_3,… // a series of one-shot IS

```
Each process p_i with input v_i:
r := 0
while true do
  r := r+1
  v_i := IS_r.WriteRead_i(v_i) // write and take a
                                  snapshot
```

IS^k represented as $\chi^k(I)$:

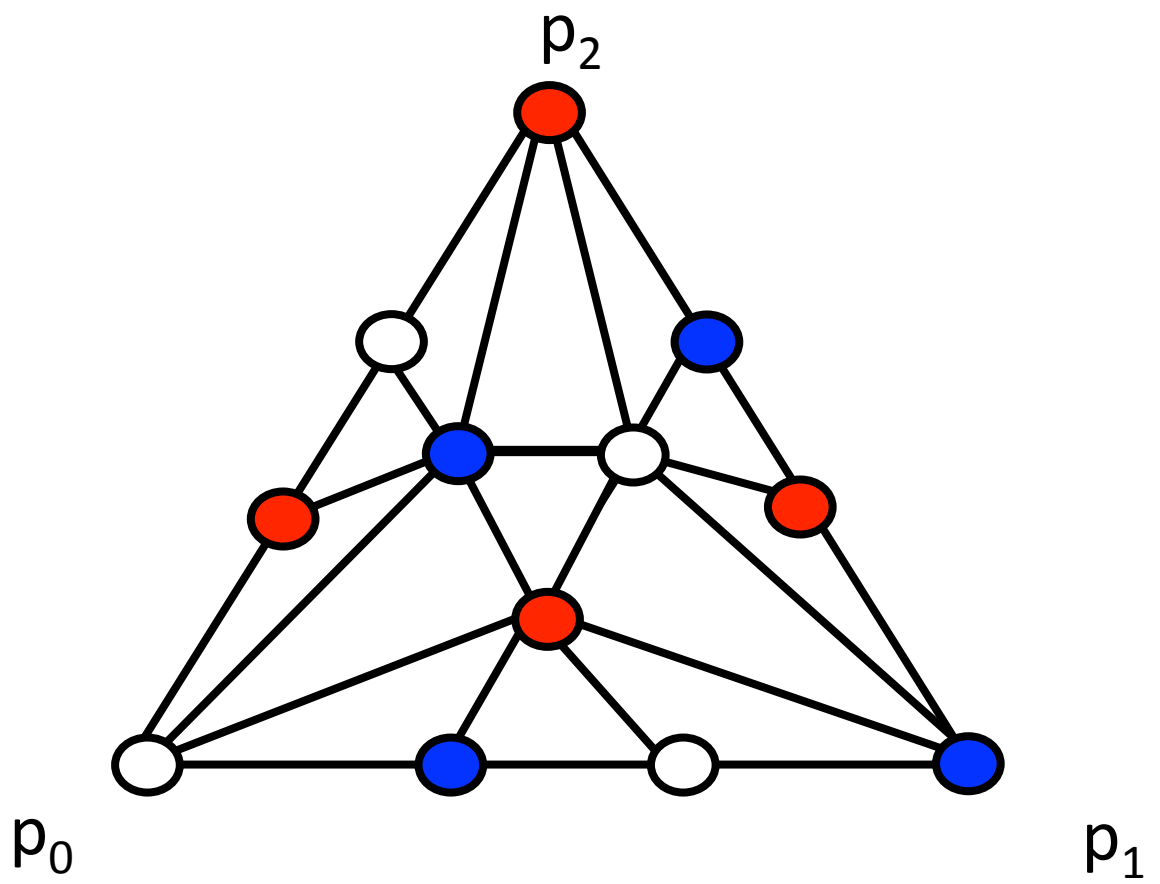
every run of IS^k ⇔ a simplex in $\chi^k(I)$

$\chi^2(\mathtt{I})$: 2-round IIS



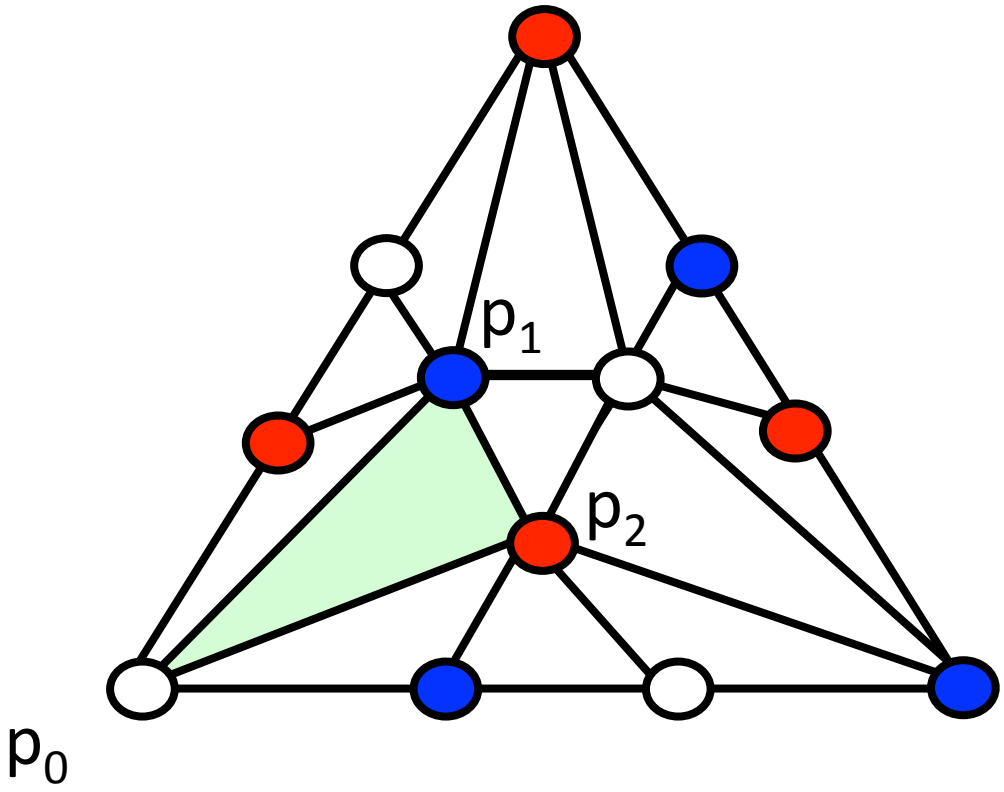
IS as a task $(s^N, \mathcal{X}(s^N), \mathcal{X})$

A process starts at its corner…



IS as a task $(s^N, \mathcal{X}(s^N), \mathcal{X})$

and outputs a vertex of it color (carrier-preserving)



Chromatic simplex agreement on $\chi(I)$

IS - the task for wait-freedom

Read-write model (RW) and IIS are equivalent [BG93,BG97,GR10]

- a task is solvable in IIS iff it is solvable in RW

Asynchronous computability theorem[HS93]:

A task (I,O,Δ) is wait-free read-write solvable if and only if there is a chromatic simplicial map from a subdivision $\chi^r(\mathrm{I})$ to O carried by Δ

Model as a task?

- M model, a set of (infinite) runs
  - ✓Alternating writes and snapshots

- T task, a one-shot distributed function (I,O,Δ):
  - ✓Set of input vectors I (input complex)
  - ✓Set of output vectors O (output complex)
  - ✓Task specification Δ: I ➔ 2^O (carrier map)

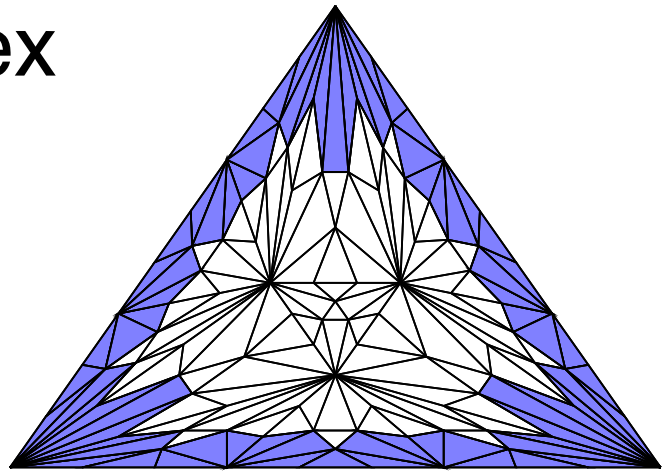
- T*, iterations of T, have the same task computability as M

(Solving a task in M is equivalent to solving T)

Affine tasks

(s^N, L, Δ):

- s^N – N-dimensional simplex
- $L \subseteq \mathcal{X}^k(s^N)$
- $\Delta(\sigma) = \mathcal{X}^k(\sigma) \cap L$



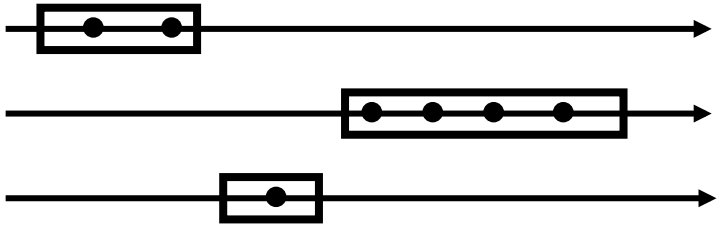
$L = \mathcal{X}^k(s^N)$: IS

Model as a task

- IS is the matching affine task for wait-free runs
  - ✓What about restrictions of wait-free?

- k-concurrency?
  - ✓a subset of RW runs where at most k process are concurrently active

Concurrency levels [Gaf09]

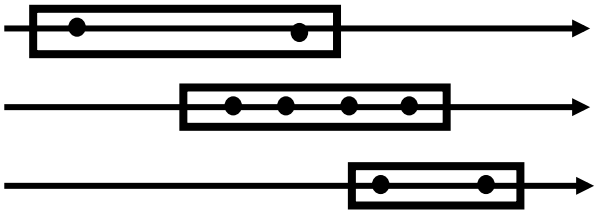


1-concurrent: at most one process makes progress at a time

(global lock)

k-concurrent: at most k processes make progress concurrently

(k-resource semaphore)



n-concurrency = wait-freedom ≅ IS

A matching affine task for k-concurrency (0<k<n)?

Defining R_k

Contention sets: all the processes that share a carrier (≈ see each other):

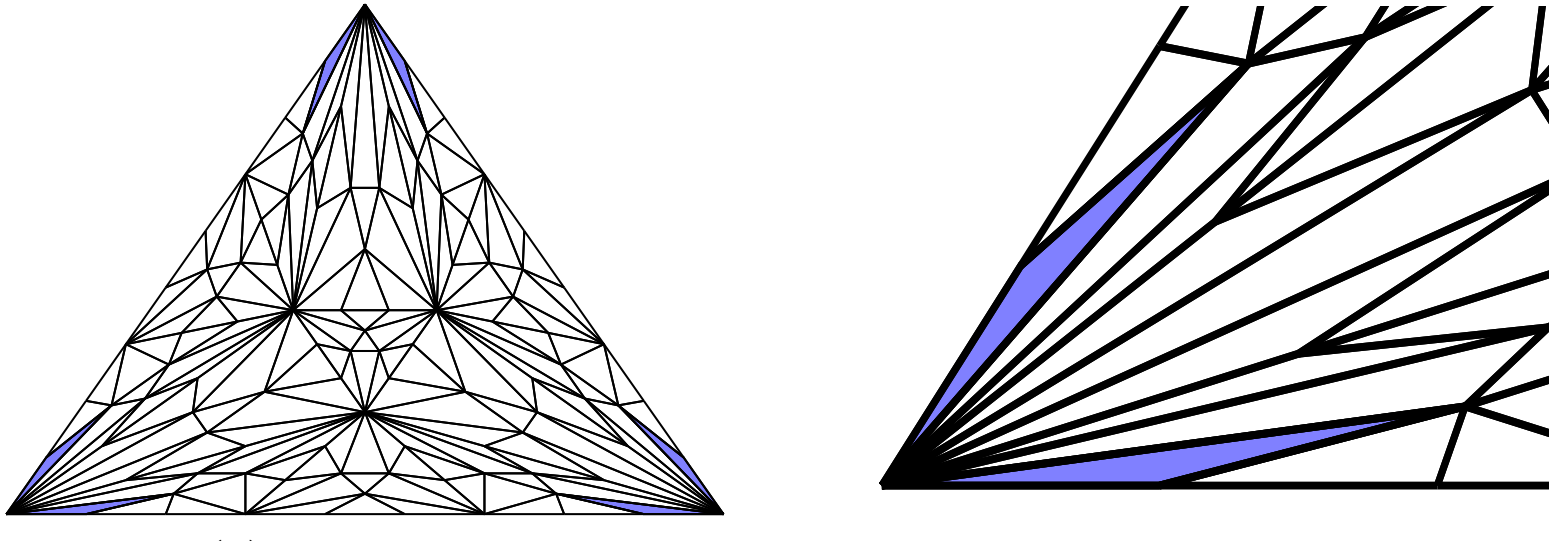
$$Cont(\sigma) = \{S \subseteq \Pi, \forall p, p' \in S, carrier(p, \sigma) = carrier(p', \sigma)\}$$

Include all simplices in $\mathcal{X}^2(s^N)$ of contention k or less

$$\mathcal{R}_k = \{\sigma \in \mathrm{Chr}^2\, \mathbf{s}, \forall S \in Cont(\sigma), |S| \leq k\}$$

R_1

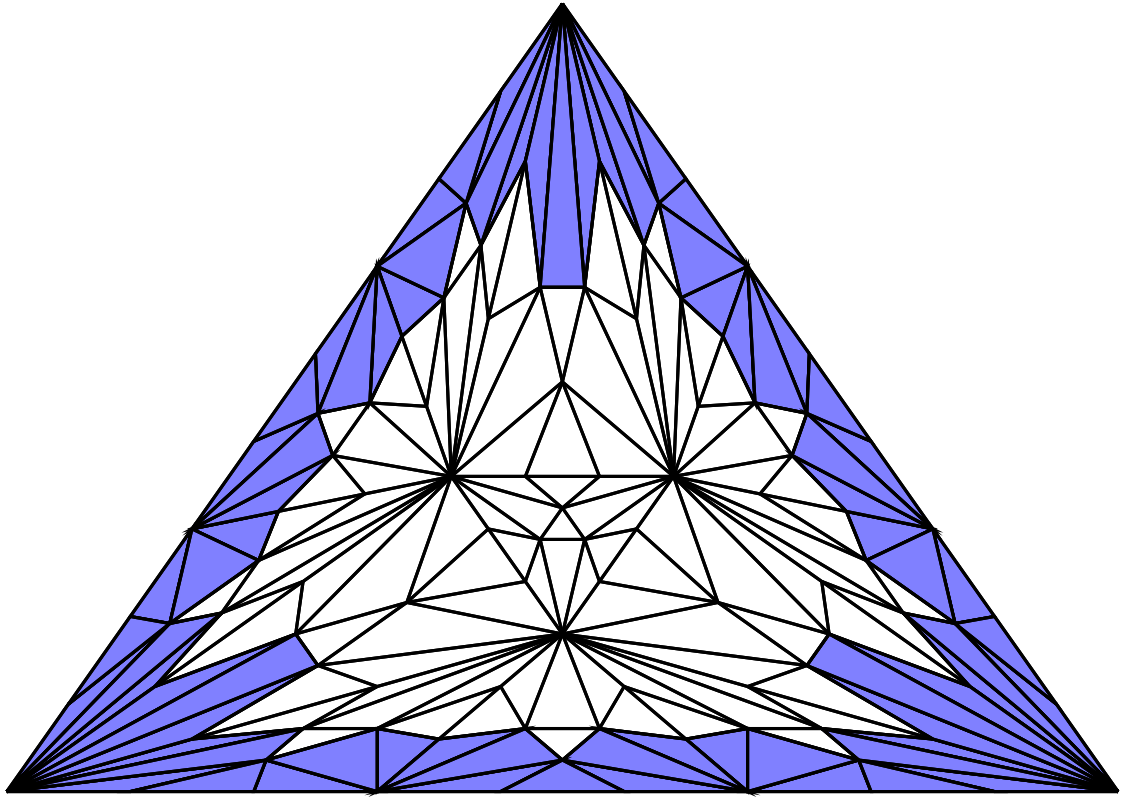
Process proceed in the same total order in two IS rounds:



L_{ord}: total order task for s^2

R_2

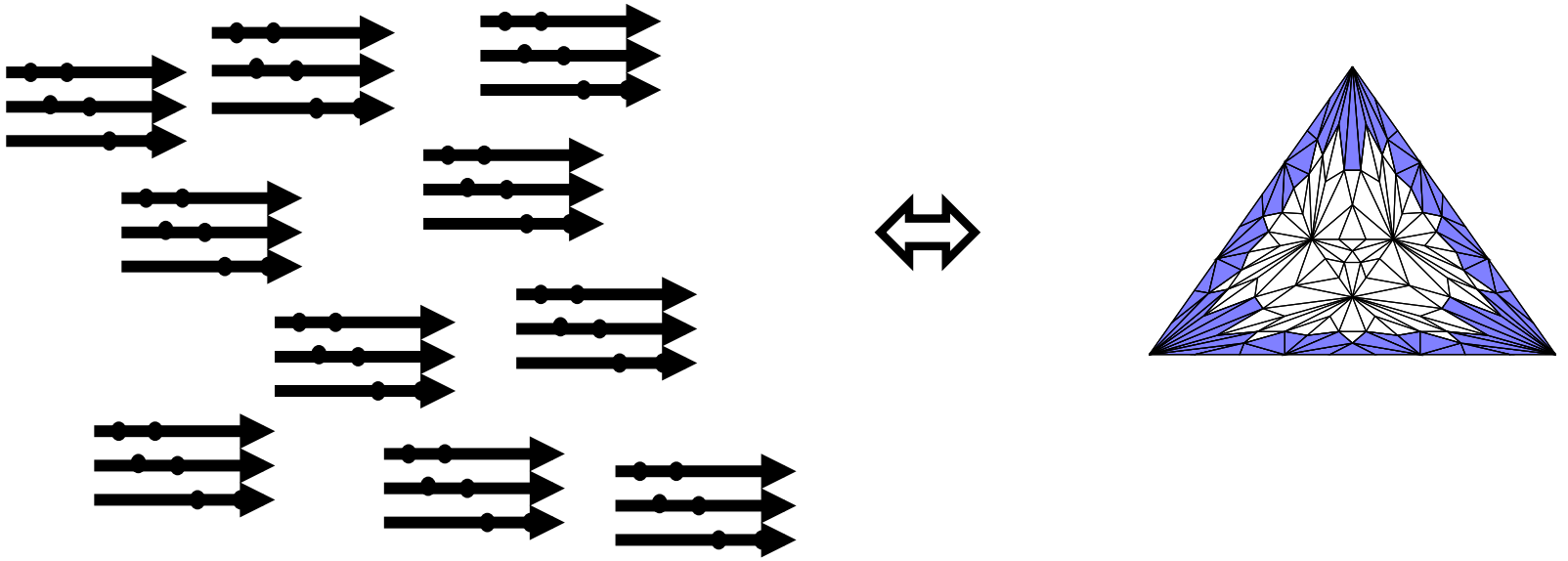
All simplices that touch 1-dimenional faces



k-concurrency = R_k*

T is solvable in R_k* iff T is solvable k-concurrently:

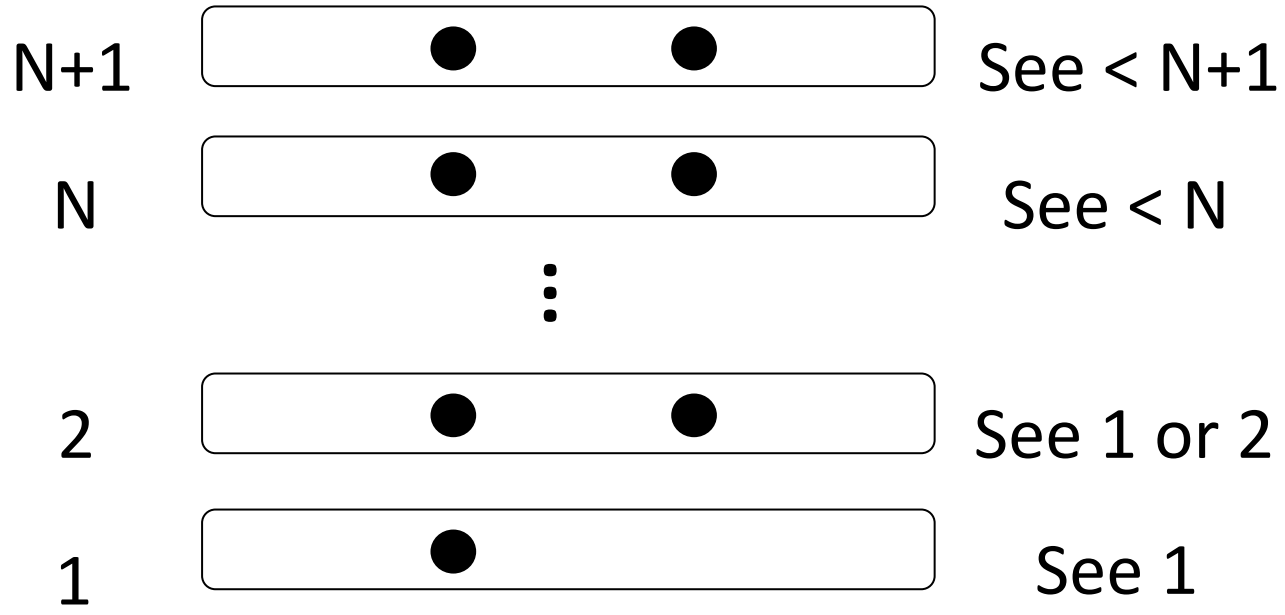
1. k-concurrency simulates R_k*
2. R_k* simulates k-concurrency



1. From k-concurrency to R_k^*

R_k can be solved k-concurrently:

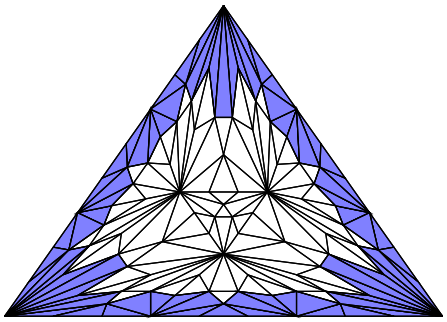
k-concurrent chromatic simplex agreement on R_k



Two rounds of k-concurrent IS implementation [BG93] give R^k

2. From to R_k^* *to* k-concurrency

- R^k can be used to solve k-set agreement:
  - ✓Decide on the value of (up to k) "leaders" processes (chosen by the size of IS^1 output)
- IIS (and thus R_k^*) can simulate RW [BG97,GR10]

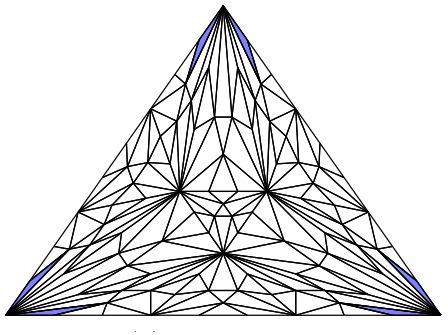


Simulate a protocol that uses read-write and k-set consensus objects?

Not that simple: how to combine simulating RW with solving k-SA?

Example: total order (k=1)

Solution of any task (I,O,Δ) in just one iteration of L_{ord}



{p0},{p1},{p2} I {p0},{p1},{p2}

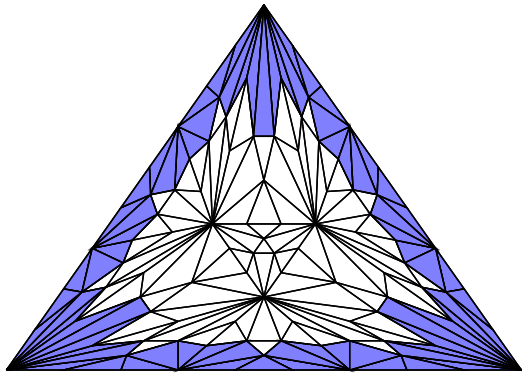
Decide on
a (deterministically chosen) value v_0 in Δ({p0})

Decide on
a value v_1 Δ({p0,p1})
In the link of v_0

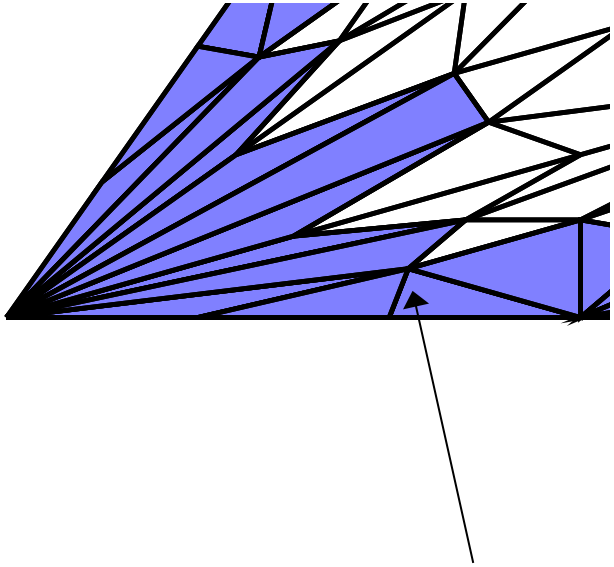
Decide on
a value v_2 in Δ({p0,p1,p2})
In the link of $\{v_0, v_1\}$

Example: R_2

More iterations might be needed



{p0},{p1},{p2} I {p0},{p1},{p2}



Who are the leaders?

Simulating k-concurrency

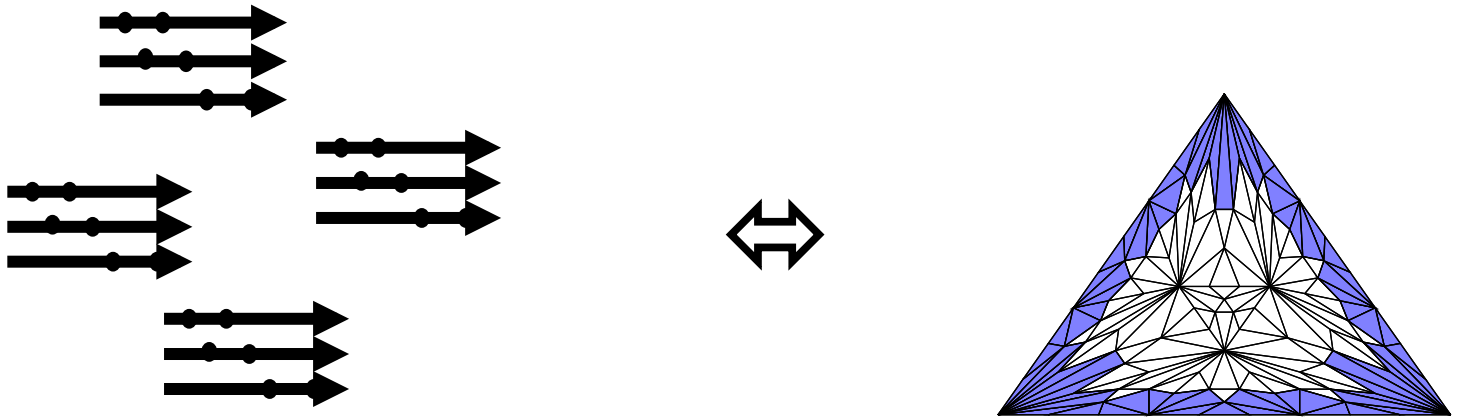
- Adaptive k-set consensus
  - ✓k-commit-adopt: commit (decide) if among k “fastest” non-terminated processes, adopt otherwise
- RW + (adaptive) k-set consensus => k state machines
  - ✓Generalized universality [GG11]
  - ✓m active simulators: machines *1..min(m,k)* are active
  - ✓Any RW protocol on up to k state machines can be simulated
- k processes simulate a k-concurrent system
  - ✓Extended BG simulation [Gaf09]
  - ✓Let state machines be (EBG) simulators

RW + k-set agreement simulate k-concurrency

k-concurrency = R_k*

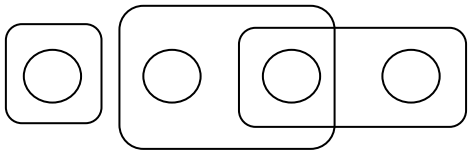
T is solvable in R_k* iff T is solvable k-concurrently:

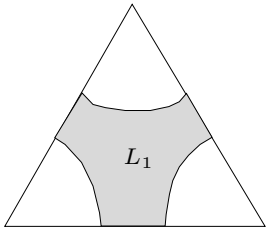
1. k-concurrency simulates R_k*
2. R_k* simulates k-concurrency



Other models?

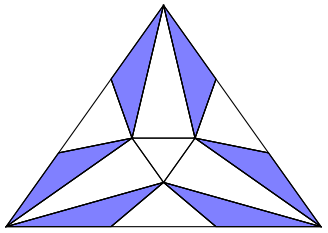
- Adversarial models [DFGT09]
  - ✓Non-uniform/correlated faults
  - ✓[SHG16]: t-resilience





- Set-consensus collections [DFGK16]
  - ✓RW + set-consensus objects in $\{(s_1,t_1),\ldots,(s_m,t_m)\}$
  - ✓k-concurrency ≅ k-set consensus

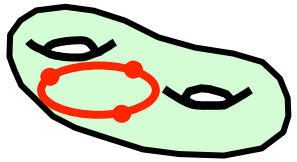
- Affine tasks are in $\mathcal{X}^2(s^N)$
  - ✓Sometimes even in $\mathcal{X}^1(s^N)$



2-consensus
(TAS)

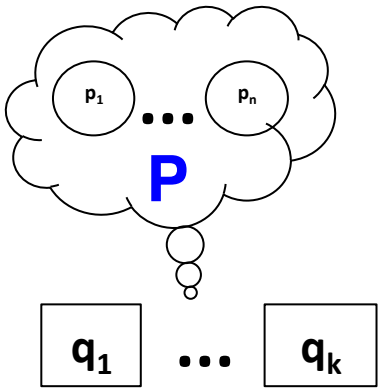
What is good about it?

- Compact representation of non-compact models
- Conjecture: possible for all “natural models”
  - ✓ Captured by computing artifacts
  - ✓ Not 0-1-exclusion, WSB, Möbius etc.
- Conjecture: relations between models (affine tasks) are decidable
  - ✓ Reduces to maps between bounded sub-complexes of $\mathcal{X}^2(s^N)$
  - ✓ 3-process, read-write wait-free solvability of (colorless) tasks are undecidable [GK95,HR97]



Tunable BG simulation

- Simulated model
  - ✓ RW [BG93, Gaf09,…]
  - ✓ RW+{set-consensus objects} [CR93,GG11,RS13, DFGK16...]
  - ✓ Restricted concurrency [FGRR14,GHKR16,...]
  - ✓ Adversarial [GK10, DFGK16]
- Simulating model
  - ✓ Static RW [BG93,Gaf09,…]
  - ✓ RW+{set-consensus objects} [...,AEG16,DFGK15]
  - ✓ Adversarial [GK10,GK11,DFGK16]
  - ✓ Message-passing Byzantine [IRS16]
- Simulated protocols
  - ✓ Colorless [BG93,RS14,GK10]
  - ✓ Generic [Gaf09,…GHKR16]
- Agreement protocols
  - ✓ Safe agreement [BG93,…]
  - ✓ Extended agreement [Gaf09,…]
  - ✓ OF consensus [Kuz13]



Xtiuzu'u!

- E. Gafni, Y. He, P. Kuznetsov, T. Rieutord. Read-Write Memory and k-Set Consensus as an Affine Task. OPODIS 2016. Full version arxiv: 1610:01423
- C. Delporte, H. Fauconnier, E. Gafni, P. Kuznetsov. Set-consensus collections are decidable. OPODIS 2016. Full version arxiv:1607:05636