Comparing DejaVu and MonPoly

Property 1 : Access

DejaVu:

```
prop access :
  Forall user . Forall file .
    access(user,file) ->
      [login(user),logout(user))
        &
      [open(file),close(file))
```

MonPoly:

```
FORALL x. FORALL y.
  access(x,y) IMPLIES
    (
      (NOT logout(x) SINCE login(x))
        AND
      (NOT close(y) SINCE open(y))
    )
```

Trace	DejaVu < 20	D 20	D 40	D 60	MonPoly	Winner
10K	0m0.882s (13b)	0m0.862s	0m0.966s	0m0.991s	0m1.194s	d
100K	0m2.057s (16b)	0m1.963s	0m2.769s	0m3.127s	6m12.897s	D
1000K	0m15.452s (19b)	0m14.138s	0m22.523s	0m31.780s	**TBD**~16h	D

Property 2 : File

DejaVu:

```
prop file : Forall f . close(f) -> Exists m . @ [open(f,m),close(f))
```

MonPoly:

```
FORALL f . close(f) IMPLIES (EXISTS m . PREVIOUS (NOT close(f) SINCE open(f,m)))
```

Trace	DejaVu < 20	D 20	D 40	D 60	MonPoly	Winner
10K	0m0.899s (13b)	0m0.877s	0m0.875s	0m0.919s	0m35.084s	D
100K	0m1.879s (17b)	0m1.900s	0m1.893s	0m2.123s	85m42.408s	D
1000K	no lower works	0m8.560s	0m9.630s	0m10.814s	**TBD**~DNF	D

Property 3 : Fifo

Original property

DejaVu:

```
prop fifo :
 Forall x .
  (enter(x) -> ! @ P enter(x)) &
  (exit(x) -> ! @ P exit(x)) &
  (exit(x) -> @ P enter(x)) &
  (Forall y . (exit(y) & P (enter(y) & @ P enter(x))) -> @ P exit(x))
```

MonPoly:

```
FORALL x.
  (
    (enter(x) IMPLIES NOT PREVIOUS ONCE enter(x)) AND
    (exit(x) IMPLIES NOT PREVIOUS ONCE  exit(x)) AND
    (exit(x) IMPLIES PREVIOUS ONCE  enter(x)) AND
    FORALL y. (
      (exit(y) AND ONCE (enter(y) AND PREVIOUS ONCE enter(x)))
        IMPLIES
      PREVIOUS ONCE exit(x)
    )
  )
```

Trace	DejaVu < 20	D 20	D 40	D 60	MonPoly	Winner
5K	0m22.708s (13b)	2m22.694s	OOM	-	2m9.448s	D
10K	4m22.872s (14b)	OOM	-	-	14m28.818s	D
100K	OOM (14b)	-	-	-	-	-
1000K	-	-	-	-	-	-

Modified property

DejaVu:

```
prop fifo:
 Forall x. Forall y.
  (exit(x) -> @(!exit(x) S enter(x)))
  &
  (@ (!exit(x) S enter(x)) -> !enter(x))
  &
  !(
    exit(x) &
    @(
      (!enter(x) & !exit(x) & !enter(y) & !exit(y)) S (
        exit(y) &
        @(
          (!enter(x) & !exit(x) & !enter(y) & !exit(y)) S (
            enter(y) &
            @(
              (!enter(x) & !exit(x) & !enter(y) & !exit(y)) S enter(x)
            )
          )
        )
      )
    )
  )
```

MonPoly:

```
FORALL x . (exit(x) IMPLIES PREVIOUS((NOT exit(x)) SINCE[0,*] enter(x)))
AND
FORALL x . ((PREVIOUS ((NOT exit(x)) SINCE[0,*] enter(x))) IMPLIES NOT enter(x))
AND
FORALL x . FORALL y .
 (NOT (exit(x) AND
      PREVIOUS (((NOT enter(x)) AND (NOT exit(x)) AND (NOT enter(y)) AND
         (NOT exit(y))) SINCE[0,*] (exit(y) AND
         PREVIOUS (((NOT enter(x)) AND (NOT exit(x)) AND (NOT enter(y)) AND
            (NOT exit(y))) SINCE[0,*] (enter(y) AND
            PREVIOUS (((NOT enter(x)) AND (NOT exit(x)) AND
                (NOT enter(y)) AND (NOT exit(y)))
                SINCE[0,*] enter(x))))))))
```

Trace	DejaVu < 20	D 20	D 40	D 60	MonPoly	Winner
5K	1m22.318s (13b)	10m41.721s	OOM	-	NOT MON	D
10K	OOM (14b)	-	-	-	NOT MON	D
100K	-	-	-	-	NOT MON	-
1000K	-	-	-	-	NOT MON	-

Property 4 : Lock properties

4-1 sleepSafely:

DejaVu:

```
prop sleepSafely :
  Forall t . Forall l .
    sleep(t) -> ![acq(t,l),rel(t,l))
```

MonPoly:

```
FORALL t. FORALL l .
  sleep(t) IMPLIES NOT (NOT rel(t,l) SINCE acq(t,l))
```

Trace	DejaVu < 20	D 20	D 40	D 60	MonPoly	Winner
10K	0m0.842s (10b)	0m0.855s	0m0.939s	0m1.168s	0m0.207s	M
100K	0m1.346s (13b)	0m1.241s	0m1.325s	0m1.422s	0m0.259s	M
1000K	0m2.508s (8b)	0m2.657s	0m2.982s	0m8.275s	0m4.701s	d

4-2 oneThread:

DejaVu:

```
prop oneThread :
  Forall t . Forall l .
    acq(t,l) -> ! exists s . @ [acq(s,l),rel(s,l))
```

MonPoly:

```
FORALL t. FORALL l .
  acq(t,l) IMPLIES NOT EXISTS s . PREVIOUS (NOT rel(s,l) SINCE acq(s,l))
```

Trace	DejaVu < 20	D 20	D 40	D 60	MonPoly	Winner
10K	0m1.089s (10b)	0m1.274s	0m1.963s	0m2.764s	0m1.653s	d
100K	0m1.539s (13b)	0m1.579s	0m1.866s	0m2.238s	0m0.248s	M
1000K	0m3.456s (8b)	0m3.371s	0m3.783s	0m8.587s	0m22.846s	D

4-3 releaseAcquired:

DejaVu:

```
prop releaseAcquired :
  Forall t . Forall l .
    rel(t,l) -> @ [acq(t,l),rel(t,l))
```

MonPoly:

```
FORALL t. FORALL l .
  rel(t,l) IMPLIES PREVIOUS (NOT rel(t,l) SINCE acq(t,l))
```

Trace	DejaVu < 20	D 20	D 40	D 60	MonPoly	Winner
10K	0m0.876s (10b)	0m0.866s	0m0.926s	0m0.916s	0m0.201s	M
100K	0m1.237s (13b)	0m1.192s	0m1.127s	0m1.378s	0m0.260s	M
1000K	0m2.687s (8b)	0m2.756s	0m2.857s	0m8.244s	0m5.336s	d

All Combined:

DejaVu:

```
prop locksBasic :
  Forall t . Forall l .
    (
      (sleep(t) -> ![acq(t,l),rel(t,l))) &
      (acq(t,l) -> ! exists s . @ [acq(s,l),rel(s,l))) &
      (rel(t,l) -> @ [acq(t,l),rel(t,l)))
    )
```

MonPoly:

```
FORALL t. FORALL l .
    (
      (sleep(t) IMPLIES NOT (NOT rel(t,l) SINCE acq(t,l))) AND
      (acq(t,l) IMPLIES NOT EXISTS s . PREVIOUS (NOT rel(s,l) SINCE acq(s,l))) AND
      (rel(t,l) IMPLIES PREVIOUS (NOT rel(t,l) SINCE acq(t,l)))
    )
```

Trace	DejaVu < 20	D 20	D 40	D 60	MonPoly	Winner
10K	0m1.164s (10b)	0m1.580s	0m3.499s	0m23.507s	0m2.053s	d
100K	0m2.114s (13b)	0m1.988s	0m2.145s	0m2.575s	0m0.408s	M
1000K	0m7.661s (8b)	0m10.209s	0m14.844s	0m20.870s	0m29.571s	D

Property 5 : Deadlocks

DejaVu:

```
prop locksDeadlocks :
  Forall t1 . Forall t2 . Forall l1 . Forall l2 .
    (@ [acq(t1,l1),rel(t1,l1)) & acq(t1,l2))
    ->
    (! @ P (@ [acq(t2,l2),rel(t2,l2)) & acq(t2,l1)))
```

MonPoly:

```
FORALL t1 . FORALL t2 . FORALL l1 . FORALL l2 .
    ((PREVIOUS (NOT rel(t1,l1) SINCE acq(t1,l1))) AND acq(t1,l2))
    IMPLIES
    (NOT PREVIOUS ONCE ((PREVIOUS (NOT rel(t2,l2) SINCE acq(t2,l2))) AND acq(t2,l1)))
```

Trace	DejaVu < 20	D 20	D 40	D 60	MonPoly	Winner
10K	0m1.843s (10b)	0m6.988s	0m27.335s	0m54.265s	0m1.120s	m
100K	0m1.548s (13b)	0m1.589s	0m1.755s	0m1.718s	0m6.918s	D
1000K	0m14.233s (7b)	0m47.698s	1m55.029s	4m1.124s	0m23.726s	d

Property 6 : Data races

DejaVu:

```
prop locksDataraces :
  Forall t1 . Forall t2 . Forall x .
    (
      (P (read(t1,x) | write(t1,x)))
      &
      (P write(t2,x))
    )
    ->
    Exists l .
      (
        H ((read(t1,x) | write(t1,x)) -> [acq(t1,l),rel(t1,l)))
        &
        H ((read(t2,x) | write(t2,x)) -> [acq(t2,l),rel(t2,l)))
      )
```

MonPoly:

```
  FORALL t1 . FORALL t2 . FORALL x .
    (
      (
        (ONCE (read(t1,x) OR write(t1,x)))
        AND
        (ONCE write(t2,x))
      )
      IMPLIES
      EXISTS l .
        (
          (PAST_ALWAYS ((read(t1,x) OR write(t1,x))
              IMPLIES (NOT rel(t1,l) SINCE acq(t1,l))))
          AND
          (PAST_ALWAYS ((read(t2,x) OR write(t2,x))
              IMPLIES (NOT rel(t2,l) SINCE acq(t2,l))))
        )
    )
```

Trace	DejaVu < 20	D 20	D 40	D 60	MonPoly	Winner
10K	0m1.030s (6b)	0m1.225s	0m1.480s	0m1.782s	NOT MON	D
100K	0m2.071s (10b)	0m2.606s	0m3.031s	0m4.471s	NOT MON	D
1000K	0m6.713s (9b)	0m6.795s	0m7.817s	0m8.527s	NOT MON	D