More on *tracks*
Notes on UNITY: 33
5/8/00

The goal of detection is for a predicate p to detect the occurrence of a predicate q. The meaning of detection is captured by the following requirements; (1) **invariant** $p \Rightarrow q$, (2) if q continues to hold then p will be established, and (3) once p holds it continues to hold until q becomes *false*. Therefore, we have p *tracks* q defined by:

> **invariant** $p \Rightarrow q$,
> $q \mapsto \neg q \vee p$,
> p **co** $\neg q \vee p$.

These are equivalent to the conditions in *Notes on UNITY: 30*. The two important results proven about tracks in that note are: (1) tracks is a partial order, (2) a predicate can be tracked by tracking each of its conjuncts independently. However, the following disjunction rule does not hold.

$$\frac{p \textit{ tracks } q,\ r \textit{ tracks } s}{p \vee r \textit{ tracks } q \vee s}$$

Asynchrony If p, q are known to change asynchronously, i.e., for all c, d
$p, q = c, d$ **co** $p = c \vee q = d$
then the conditions for tracks can be simplified, as shown below. Let us write p *lags* q to stand for:

> **invariant** $p \Rightarrow q$,
> $q \mapsto \neg q \vee p$,
> **stable** p.

Theorem Given the asynchrony condition, tracks and lags are equivalent.

It is easy to see that the lags implies tracks, by weakening the rhs of **stable** p. Now, we show that tracks and the asynchrony condition imply **stable** p in lags..

$p \wedge q$ **co** $p \vee q$	, setting $c, d = \textit{true}, \textit{true}$ in the asynchrony condition
p **co** $\neg q \vee p$	, from the definition of tracks
$p \wedge q$ **co** p	, conjunction of the above two
p **co** p	, using **invariant** $p \Rightarrow q$

The intuitive explanation for why p has to be stable is as follows. Given $p \Rightarrow q$, the falsification of q has to be synchronously accompanied by the falsification of p; since this is impossible, q should not be falsified if p holds, i.e., p **co** q. Conjoining with p **co** $\neg q \vee p$, we have **stable** p. Observe that q need not be stable; q may be falsified as long as p has not been established.

We can show that *lags* is antisymmetric and transitive (it may not be reflexive since p may not be stable). The proof of *leads-to* in transitivity is as follows; given p *lags* q and q *lags* r:

$q \mapsto \neg q \vee p$	, from p *lags* q
stable q	, from q *lags* r
$q \mapsto q \wedge p$	, PSP
$q \mapsto p$	, from p *lags* q, we have $p \Rightarrow q$
$r \mapsto \neg r \vee q$	, from q *lags* r
$r \mapsto \neg r \vee p$	, cancellation on the above two

The relation *lags* satisfies theorem 2 of Unity Notes 30. It also satisfies theorem 3, because *lags* is stronger than *tracks*, and *tracks* satisfies theorem 3. The disjunction result does not hold; see the counterexample below, where b is boolean and x is integer.

initially $b, x = \mathit{true}, 2$
α:: $b \wedge x = 2 \rightarrow x := 0$
β:: $\neg b \wedge x = 2 \rightarrow x := 1$
γ:: $x = 2 \rightarrow b := \neg b$

It can be shown that $x = 0$ *lags* b and $x = 1$ *lags* $\neg b$. However, $x = 0 \vee x = 1$ *lags true* does not hold. In particular, $\mathit{true} \mapsto x = 0 \vee x = 1$ does not hold; consider $(\beta\gamma\alpha\gamma)^\omega$.

An even Stronger Definition It is clear that asynchronous detection is possible only if q is stable. It is then seen that p is stable. We restrict attention to stable predicates only. Then we define *detects*, as before. All the given theorems hold. Additionally, disjunction condition holds.