Storm 计算可靠性相关引理及定理证明

符号及其对应含义如下表：

符号	含义	符号	含义
sp	输出组件	bt	处理组件
cid	组件中的标识	*id*	组件读入或发送的元组标识
mid	输入组件发送的元组（原始元组）标识	*sid*	元组流的标识
TE	迁移事件集合	*RE*	storm 拓扑运行的事件集合
stp	storm 计算拓扑	*M*	*Net* 库所标识映射 (Marking)
Net	storm 计算拓扑对应 CPN 模型	*b*	*Net* 迁移中变量的赋值
t	*Net* 迁移发生的时间	$[\ ]: var \to \Sigma$	赋值映射

Storm *Net* 中的迁移触发时将产生迁移事件，其定义如下：

定义 1（迁移事件）迁移事件 $te =< tr,b,t >$，其中：

- $tr \in TR$ 是迁移；
- $b:V \to \Sigma$ 是迁移发生时的状态，即对相关变量的赋值；
- $t \in TIME$ 表示迁移 tr 发生在时刻 t 。

$te =< tr,b,t > \in TE$ 表示在时刻 t 和状态 b 条件下发生了迁移 tr 。符号 *TE* 表示所有的迁移事件集合。

定义 2 （可执行的迁移事件 te ）给定 Marking M，时间 t^* 及迁移事件 te，te 在 $< M,t^* >$ 条件下可发生迁移的，记为 $< M,t^* > \xrightarrow{te} < M',t' >$，当且仅当以下条件是满足的：

- $t' \geq t^*$；
- 对迁移 tr 的所有库所 p, $E(p,tr)\langle b\rangle \leq M(p)$，即输入库所存在足够的托肯供迁移 tr 消耗。
- 当迁移事件 te 在 $< M,t^* >$ 下是可执行的，则 tr 发生后，Marking M' 可以定义为：

$M'(p) = (M(p) - -E(p,tr)\langle b\rangle_{+t}) + +E(tr,p) < b >_{+t}$。

定义 3 （Storm 的迁移事件）$TE = TE(sp) \cup TE(bt) \cup \overline{sAck} \cup \overline{sFail}$ 表示 *Net* 迁移事件集合，其中：

— $TE(sp) = \overline{take(sp)} \cup \overline{semit(sp)}$，表示输出组件对应的迁移事件集合；

— $TE(bt) = \overline{take(bt)} \cup \overline{emit(bt)} \cup \overline{ack(bt)} \cup \overline{fail(bt)}$，表示处理组件对应的迁移事件集合，由 *take*，*emit*，*ack*，*fail* 类型迁移构成。

— $\overline{sAck}=\{(sAck,b,t)\}$，表示计算确认迁移事件集合；

— $\overline{sFail}=\{(sFail,b,t)\}$，表示计算失败迁移事件集合；

定义 4（Storm 的运行事件）$RE=RE(sp)\bigcup RE(bt)\bigcup RE(ack)$ 表示 stp 运行时日志事件集合，其中：

— $RE(sp)=\{semit(cid,mid,sid,t)\}$，表示标识为 *cid* 的输出组件在 *t* 时刻发送标识为 *mid* 的元组至元组流 *sid* 中。

— $RE(bt)=\{take(cid,id,mid,sid,t)\}\bigcup\{emit(cid,id,mid,sid,t)\}\bigcup$
$\{fail(cid,id,mid,t)\}\bigcup\{ack(cid,id,mid,t)\}$；

— $RE(ack)=\{sAck(mid,t)\}\bigcup\{sFail(mid,t)\}$。

RE 和 *TE* 两类事件中间存在的映射关系由下述定义确定。

定义 5（*RE* 到 *TE* 的映射）给定 Storm 拓扑 *stp*，*RE* 事件集到 *TE* 迁移事件集的映射关系由映射函数 $rt:RE\to\overline{TE}$，通过如下规则定义：

—当 $e=semit(cid,mid,sid,t)\in RE(sp)$，其中设 v_0 为输出组件 sp 读入的数据，tup_0 为发送的元组，则 $rt(e)=tk\cdot se$，其中

- $tk=(take(cid),[v\mapsto v_0],t^-)\in\overline{take(sp)}$；
- $se=(semit(cid,so_i),[tup\mapsto tup_0],t\ )\in\overline{semit(sp)},(so_i.sid=sid)$；

—当 $e=take(cid,id,mid,sid,t)\in RE(bt)$，其中设 tup_0 为组件 bt 从元组流 *sid* 中读取的元组 $(tup_0.id=id,\ tup_0.mid=mid)$，则 $rt(e)=tk$，其中 $tk=(take(cid),[tup\mapsto tup_0],t)\in\overline{take(bt)}$；

—当 $e=emit(cid,id,mid,sid,t)\in RE(bt)$，设 tup_0 为组件 bt 向元组流 *sid* 中发送的元组 $(tup_0.id=id,\ tup_0.mid=mid)$，则 $rt(e)=et$，其中 $et=(emit(cid,so_i),[tup\mapsto tup_0],t)\in\overline{emit(bt)}$；

—当 $e=fail(cid,id,mid,t)\in RE(bt)$，则 $rt(e)=fa$，其中 $fa=(fail(cid),[re\mapsto(cid,mid,FALSE)],t)\in\overline{fail(bt)}$；

—当 $e=ack(cid,id,mid,t)\in RE(bt)$，则 $rt(e)=(ack(cid),[re\mapsto(id,mid,TRUE)],t)\in\overline{ack(bt)}$；

—当 $e=sAck(mid,t)$，则 $rt(e)=(sAck,[re\mapsto(mid,TRUE)],t)\in\overline{sAck}$，其中 re 为所有处理组件中的变量；

—当 $e=sFail(mid,t)$，则 $rt(e)=(sFail,[re_0\mapsto(mid,FALSE)],t)\in\overline{sFail}$，其中 re_0 为某个调用 *fail* 方法的组件中的结果变量。

设 $\bar{\sigma}\in\overline{RE}$ 为运行迹，$\overline{rt}(\bar{\sigma})\in\overline{TE}$ 定义为：若 $\bar{\sigma}=\bar{\sigma}'\cdot e$，则 $\overline{rt}(\bar{\sigma})=\overline{rt}(\bar{\sigma}')\cdot rt(e)$。

Storm 拓扑 *stp* 运行迹与对应 *Net* 的迁移迹满足如下关系：

引理 1 设 Storm 拓扑 *stp*，及对应的 *Net*，如果 $\bar{\sigma}$ 为 *stp* 关于输入数据 v 的完整运行迹，则 $\overline{rt}(\bar{\sigma})$ 为

Net 关于 v 的完整迁移迹。

证明：

（1） $\overline{rt}(\bar{\sigma})$ 为 *Net* 关于 v 的迁移迹。

对 $\bar{\sigma}$ 的长度 $n=|\bar{\sigma}|$ 进行归纳假设。

当 n=1 时，根据 Storm 拓扑的执行语义和事件打点方式可知，此时 $\bar{\sigma} \in semit(cid,mid,sid,t)$ 表示 Storm 首先在输出组件 cid 中，在 *t* 时刻完成发送元组 *mid* 至元组流 *sid* 中。$\overline{rt}(\bar{\sigma}) = tk \cdot se \in \overline{TE}$，显然满足，$< M_0,0 > \xrightarrow{tk \cdot se} < M,t >$，其中 $M_0(p_0)=v$，$M(p_{so_i})=tup$，$(so_i.id=sid, tup.id=mid)$ 即 $\overline{rt}(\bar{\sigma})$ 为 *Net* 的迁移迹。

设 $|\bar{\sigma}| \le n$ 时上述命题成立，则当 $|\bar{\sigma}|=n+1$ 时，令 $\bar{\sigma}[n]=e$，$\bar{\sigma}[n+1]=e'$。

由假设可知，$\bar{\sigma}[0:n]$ 满足上述命题，则存在 M_0，t^-，M'，使得 $< M_0,0 > \xrightarrow{\overline{rt}(\bar{\sigma}[0:n])} < M^-,t^- >$。下面须证明，存在 *Net* 的库所标识 M'，使得

$$< M^-,t^- > \xrightarrow{rt(e')} (M',t)。$$

对 e， e' 之间的关系进行讨论。

a) 当 $e=semit(cid,mid,sid,t)$ 时，由于 $\bar{\sigma}$ 为处理输入 v 的运行迹。则 $e'=semit(cid,mid',sid',t')$ 或 $e'=take(cid',id,mid,sid,t')$，即同一输出组件发送其他元组至其他元组流中，或后续组件消费订阅的元组。根据迁移事件 $rt(e)$ 的定义可知，

$$< M^-,t^- > \xrightarrow{\overline{rt}(e')} (M',t)，\ M' 满足：$$

当 $e'=semit(cid,mid',sid',t')$ 时，$M'(p_{so_i'})=tup'$，其中，$so_i'.sid=sid'$，$tup'.id=mid'$。

当 $e'=take(cid',id,mid,sid,t')$ 时，$M'(p_{so_i'})=\varnothing$[1]，$M'(p_{ready}^{cid'})=\varnothing$，$M'(p_{val}^{cid'})=val(tup)$，$M'(p_{re}^{cid'})=res(tup)$。其中，$tup.id=mid$，$tup$ 为组件 cid' 读入的元组。对其他 $p \in Net$，$M'(p)=M^-(p)$。从而可知，$< M_0,0 > \xrightarrow{\overline{rt}(\bar{\sigma}[0:n])} < M^-,t^- >$ 即 $\overline{rt}(\bar{\sigma})$ 为 *Net* 的迁移迹。

b) 当 $e=take(cid,id,mid,sid,t)$ 时，根据 *id, cid, mid* 的关联性，e' 可分成两类：即

(b.1) e 与 e' 相互独立。即 $e.id \neq e'.id$ 且 $e.cid \neq e'.cid$，则设 $\bar{\sigma}'=\bar{\sigma}[0:n-1] \cdot e' \cdot e$，即交换 e 和 e' 顺序，则 $\bar{\sigma}'$ 与 $\bar{\sigma}[0:n-1] \cdot e'$ 仍为 *stp* 的运行迹，且 $|\bar{\sigma}[0:n-1] \cdot e'|=n$，此时，根据归纳假设，$rt(\bar{\sigma}[0:n-1] \cdot e')$ 为 *Net* 的迁移迹，且满足

$$< M_0,0 > \xrightarrow{rt(\bar{\sigma}[0:n-1] \cdot e')} < M^-,t^- >。$$

由于 e 与 e' 的独立性，则 $rt(e)$ 与 $rt(e')$ 也相互独立，故 M^- 满足 $< M^-,t^- > \xrightarrow{rt(e)} < M',t >$。因此 $\bar{\sigma}'$ 与 $\bar{\sigma}$ 都满足 $< M_0,0 > \xrightarrow{\overline{rt}(\bar{\sigma})} < M',t >$，$< M_0,0 > \xrightarrow{\overline{rt}(\bar{\sigma}')} < M',t >$。即 $\overline{rt}(\bar{\sigma}')$ 为 *Net* 的迁移迹。

(b.2) e 与 e' 相关联。若 $e.id=e'.id$ 且 $e.cid=e'.cid$ 时，则 $e'=fail(cid,id,mid,t)$ 或 $ack(cid,id,mid,t)$。根据 *val* 和 *res* 函数的定义可知：

---

[1] $M'(p_{so_i'})=\varnothing$ 表示库所无托肯

$$< M^{-}, t^{-} > \xrightarrow{rt(e')} < M', t >,$$

若 $e.id \neq e'.id$，$e.cid = e'.cid$，$e.mid = e'.mid$，则 $e' = emit(cid, id', mid, t)$。则根据 *take* 迁移中 *val* 函数的定义可知：

$$< M^{-}, t^{-} > \xrightarrow{rt(e')} < M', t >$$

因此，本情形下，（1）成立。

c) 其他情形可以通过同样类似分析，得到相同结论。

（2）$\overline{rt}(\bar{\sigma})$ 是完整的迁移迹。

由于 *Net* 模型中不存在其他内部迁移事件，当 $\bar{\sigma}$ 为完整的运行迹时，即不存在更多的运行，故 *Net* 不存在更多的迁移事件，即 $\overline{rt}(\bar{\sigma})$ 是完整的迁移迹。

综上所述，引理 1 成立。

引理 2（Storm 计算可靠性与运行迹）给定 Storm 计算拓扑 *stp*，若 *stp* 是计算可靠的，则对于任意 *stp* 的运行迹 $\bar{\sigma}$，$\bar{\sigma}$ 满足如下条件：

- （停机）对于 $\bar{\sigma}$ 中任意 *semit* 事件，都有以 *mid* 为关联的 *sFail* 或 *sAck* 事件相对应；
- （正确停机）若 *sAck* 事件出现在 $\bar{\sigma}$ 中，则必为 $\bar{\sigma}$ 中的最后事件；
- （bolt 处理正确）$\bar{\sigma}$ 中每个 *take* 事件都有以 *mid* 为关联的 *ack* 或 *fail* 事件；
- （通讯正常）若 $\bar{\sigma}$ 中未包含 *sFail* 事件，则 $\bar{\sigma}$ 中每个 *emit* 事件都有和其流依赖的组件的 *take* 事件对应。

证明：使用反证法证明，分为以下四步：

设 $\bar{\sigma}$ 为 *stp* 关于数据 v 的完整运行迹。

（1）$\bar{\sigma}$ 满足停机条件。

假设 $\bar{\sigma}$ 中存在 *semit* 事件，但不存在对应的 *sFail* 及 *sAck*。则根据引理 1，*Net* 存在完整运行迹 $\overline{rt}(\bar{\sigma})$。由于 *sFail*，*sAck* 事件未出现在 $\bar{\sigma}$ 中，则 $\overline{rt}(\bar{\sigma})$ 中不存在 *sFail*，*sAck* 类型迁移。故 *Net* 中的 *sAck*，*sFail* 满足 $M_f(sAck) = M_f(sFail) = \varnothing$，从而 *Net* 关于 v 的完整迁移迹是不可靠的。矛盾，故（1）成立。

（2）$\bar{\sigma}$ 满足正确停机条件。

假设 *sAck* 事件出现在 $\bar{\sigma}$ 中，但不是最后的事件。不妨设最后事件为 e。根据引理 1 可知，*Net* 存在关于数据 v 的完整迁移迹 $\overline{rt}(\bar{\sigma})$。其中最后迁移为 $rt(e) = te$。te 不是 *sAck*，*sFail* 迁移。当 te 为其他任何迁移时，必存在库所 p' $(p' \notin \{sAck, sFail\})$，使得 $M_f(P) \neq \varnothing$，显然，此时 $\overline{rt}(\bar{\sigma})$ 对 v 的计算是不可靠的。矛盾，故（2）成立。

（3）$\bar{\sigma}$ 满足每个 bolt 正确处理条件。

假设 $\bar{\sigma}$ 中存在一个 *cid* 的 bolt，其中 *take* 事件没有对应的 *ack* 和 *fail* 事件。则根据引理 1 可知，*Net* 存在完整运行迹 $\sigma = \overline{rt}(\bar{\sigma})$，满足 *take*(*cid*)迁移发生，而 *ack*(*cid*)，*fail*(*cid*)迁移未发生。则由 bolt 的语义可知，$M_f(ready_{cid}) = \varnothing$，从而 σ 不是 *Net* 可靠的运行迹。矛盾，因此（3）成立。

（4）$\bar{\sigma}$ 满足通讯正常条件

假设 $\bar{\sigma}$ 中存在某个 *emit* 事件 e，不存在后续关联的 *take* 事件 *tk*，不妨设 $e = semit(cid, id, mid, sid, t)$，$tk = take(cid', id, mid, sid, t')$。根据引理 1 可知，存在 *Net* 迁移迹 σ，

满足 $rt(e) \in \sigma$，且 σ 中不存在 $take(cid')$ 迁移。由于在处理组件中无法发生对元组 *id* 触发 *ack* 或 *fail* 迁移，则在未发生 *sFail* 迁移的前提下 $M_f(P_{ack_{cid}}) = M_f(P_{fail_{cid}}) = \varnothing$，从而 σ 不是计算可靠的迁移迹。矛盾，从而（4）成立。

综上所述，引理 2 成立。

定理 1 给定 Storm 计算拓扑 *stp*，若 *stp* 的计算是可靠的，则对任意 *stp* 的运行迹 $\overline{\sigma}$，$\overline{\sigma}$ 满足

$$\overline{\sigma} \vDash ECL(stp)$$ 。

证明：根据 ECL 公式的语义及引理 2，可证明该定理。