Mondex, elektronická peněženka

Miroslav Kovář

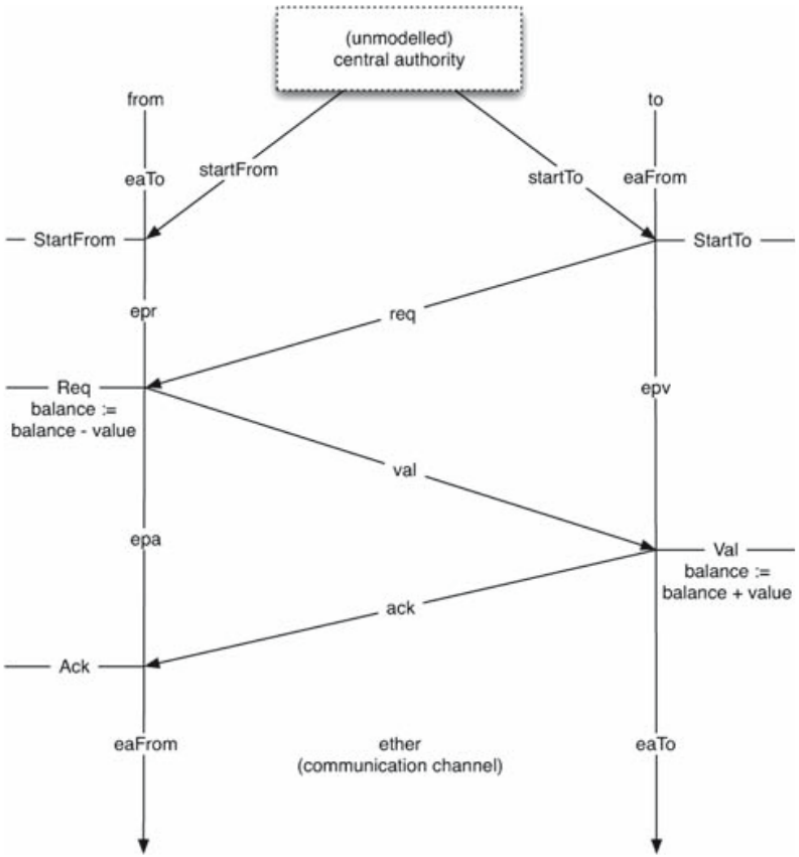
ČVUT

kovarm23@fjfi.cvut.cz

December 6, 2017

Přehled

Vymezení problému



1. Odesilatel odešle inicializační zprávu.
2. Příjemce přijme inicializační zprávu a odešle požadavek.
3. Odesilatel přijme požadavek, sníží svůj zůstatek a odešle hodnotu.
4. Příjemce přijme hodnotu, zvýší svůj zůstatek a odešle potvrzení.
5. Odesilatel přijme potvrzení.

Co se může pokazit

- Peněženka se může odpojit příliš brzy
- Zpráva se může ztratit v komunikačním kanálu
- Zpráva může být poslána vícekrát
- Můžou se vyskytnout padělané zprávy
- ...

Reprezentace mincí

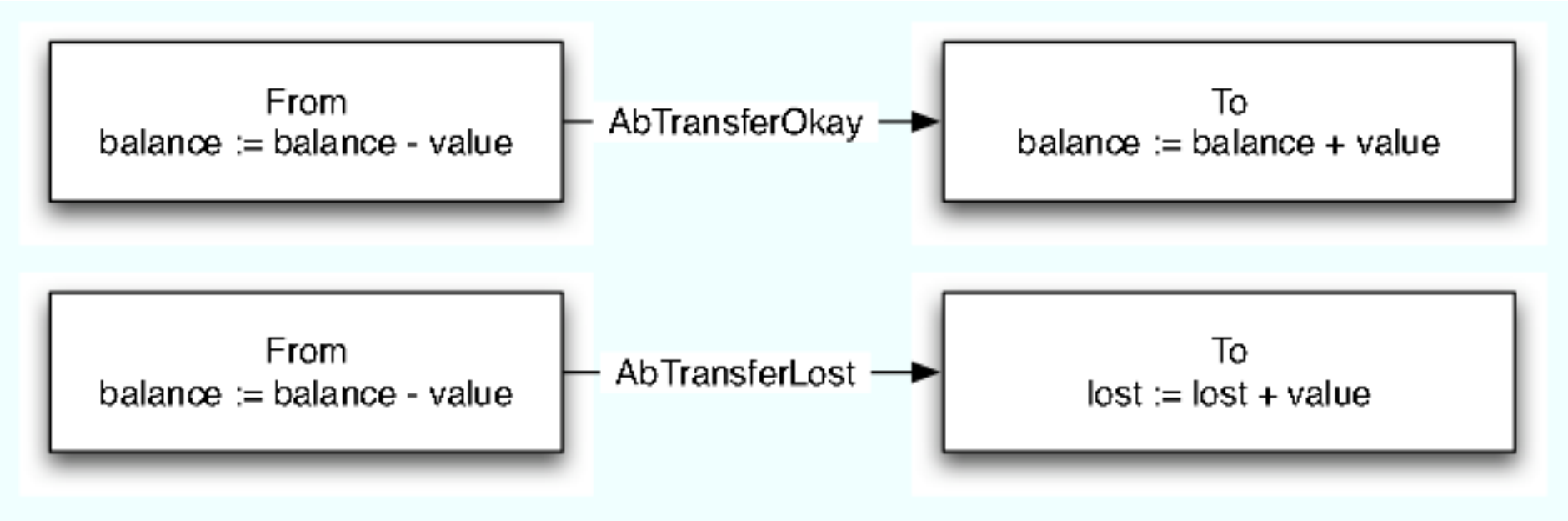
```
sig NAME {} -- Jmeno uzivatele
sig Coin {}
sig PayDetails {
  from, to: NAME,
  fromSeqNo, toSeqNo: SEQNO,
  value: set Coin
}
```

Součet je disjunktní sjednocení dvou množin.

Rozdíl je množinový rozdíl s přidanou podmínkou, že odečítaná množina musí být podmnožinou množiny, od které se odečítá.

Porovnání je inkluze množin.

Abstraktní model



Abstraktní model

```
sig AbPurse {
  balance: set Coin,
  lost: set Coin -- Mince zalogovane jako ztracene
}
sig AbWorld { abAuthPurse: NAME -> AbPurse }
fact noCoinSharing {
  all w: AbWorld {
    no disj n1, n2: NAME {
      some n1.(w.abAuthPurse).(balance + lost) &
        n2.(w.abAuthPurse).(balance + lost) -- 1
    }
    no p: AbPurse {
      p in NAME.(w.abAuthPurse)
      some p.balance & p.lost -- 2
    }
  }
}
```

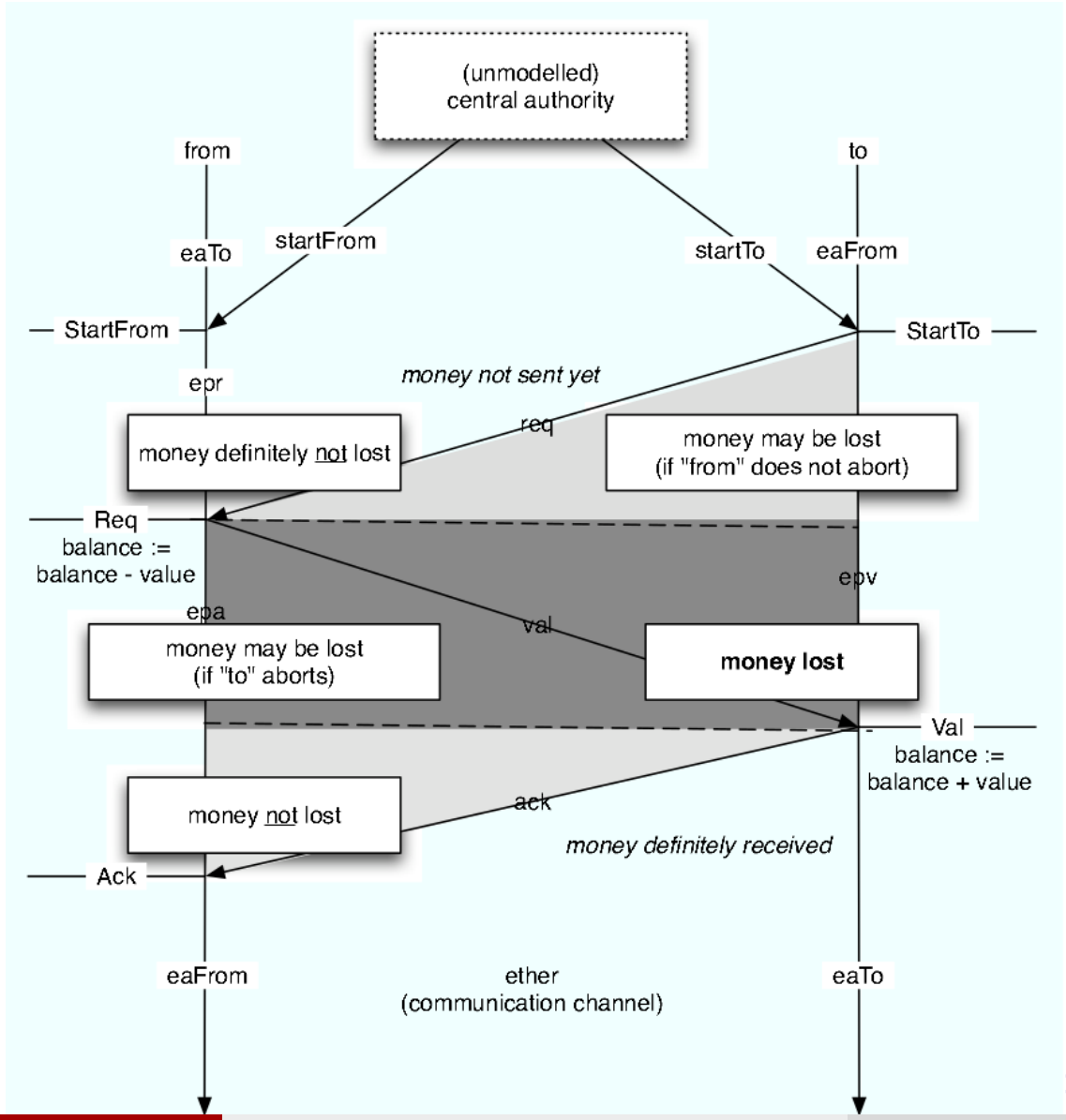
Konkrétní model: definice

```
sig ConPurse {
  name: NAME,
  balance: set Coin,
  pdAuth: PayDetails, -- Aktualni transakce - nejvyse 1
  exLog: set PayDetails, -- Lok. log nevyd. trans.
  nextSeqNo: SEQNO, -- Cislo dalsi transakce
  status: STATUS -- Status penezenky, viz dalsi obr.
}
sig ConWorld {
  conAuthPurse: NAME -> lone ConPurse,
  ether: set MESSAGE, -- Komunikacni kanal
  archive: NAME -> PayDetails -- Globalni archiv nev. trans.
}
```

Konkrétní model: podmínky

```
fact noCoinSharingConcrete {
  all p: ConPurse {
    no p.exLog.value & p.balance -- 1
  }
  all w: ConWorld {
    no disj n1, n2: NAME {
       some n1.(w.conAuthPurse).balance &
         n2.(w.conAuthPurse).balance -- 2
    }
    no p: ConPurse, pd: PayDetails {
      p in NAME.(w.conAuthPurse)
      pd in NAME.archive
      some p.balance & pd.value -- 3
    }
  }
}
```

Problémy


(unmodelled)
central authority
from
to
eaTo
startFrom
startTo
eaFrom
StartFrom
StartTo
epr
money not sent yet
req
money definitely not lost
money may be lost
(if "from" does not abort)
Req
balance :=
balance - value
epv
epa
val
money may be lost
(if "to" aborts)
money lost
Val
balance :=
balance + value
money not lost
ack
money definitely received
Ack
eaFrom
ether
(communication channel)
eaTo

Řešení

```
fun allLogs (c : ConWorld) : ConPurse -> PayDetails {
  c.archive + (c.conAuthPurse <: exLog.c)
}
fun authenticFrom (c : ConWorld) : set PayDetails {
  from.(c.conAuthPurse)
}
fun authenticTo (c : ConWorld) : set PayDetails {
  to.(c.conAuthPurse)
}
fun fromLogged (c : ConWorld) : set PayDetails {
  authenticFrom (c) & ConPurse.(allLogs (c) & ~from)
}
fun toLogged (c : ConWorld) : set PayDetails {
  authenticTo (c) & ConPurse.(allLogs (c) & ~to)
}
fun toInEpv (c : ConWorld) : set PayDetails {
  authenticTo (c) & to.status.c.epv & (iden & to.(pdAuth.c)).PayDetails
}
fun fromInEpr (c : ConWorld) : set PayDetails {
  authenticFrom (c) & from.status.c.epr & (iden & from.(pdAuth.c)).PayDetails
}
fun fromInEpa (c : ConWorld) : set PayDetails {
  authenticFrom (c) & from.status.c.epa & (iden & from.(pdAuth.c)).PayDetails
}
```

Řešení

```
fun definitelyLost (c : ConWorld) : set PayDetails {
  toLogged (c) & (fromLogged (c) + fromInEpa (c))
}

fun maybeLost (c : ConWorld) : set PayDetails {
  (fromInEpa (c) + fromLogged (c)) & toInEpv (c)
}
```

Řešení

```
all w: ConWorld {
  no p: ConPurse {
    p in NAME.(w.conAuthPurse)
    some p.balance & (definitelyLost (w) + maybeLost (w))
      -- new 3
  }
}
```

Reference

Tahina Ramanandro (2007)

Mondex, an electronic purse: specification and refinement checks with the Alloy model-finding method

Formal aspects of computing 12(3), 21 – 39.

The End