In sections:

- **Section 20.4, Syntax Box 20-2.**
- **Section 20.4.1, Syntax Box 20-3**
- **Annex A.2.11**

Change the following production (remove extra parenthesis)

FROM:

| [**wildcard**] bins_keyword bin_identifier [**[**]] **=** (trans_list) [**iff (** expression **)**]

TO

| [**wildcard**] bins_keyword bin_identifier [**[**]] **=** trans_list [**iff (** expression **)**]

As shown below:

```
cover_point ::=                                                    // from Annex A.2.11
        [ cover_point_identifer : ] coverpoint expression [ iff ( expression ) ] bins_or_empty

bins_or_empty ::=
        { {attribute_instance} { bins_or_options ; } }
      | ;

bins_or_options ::=
        coverage_option
      | [ wildcard ] bins_keyword bin_identifier [ [ [ expression ] ] ] = { range_list } [ iff ( expression ) ]
      | [ wildcard ] bins_keyword bin_identifier [ [ ] ] = trans_list [ iff ( expression ) ]
      | bins_keyword bin_identifier [ [ [ expression ] ] ] default [ iff ( expression ) ]
      | bins_keyword bin_identifier default sequence [ iff ( expression ) ]

bins_keyword::= bins | illegal_bins | ignore_bins

range_list ::= value_range { , value_range }

value_range ::=                                                    // from Annex A.8.3
        expression
      | [ expression : expression ]
```

Syntax 20-2—coverpoint syntax (excerpt from Annex A)

In sections:

- **Section 20.4.1, Syntax Box 20-3**
- **Annex A.2.11**

Change the following two productions

FROM:

trans_list ::= trans_set { , trans_set }

TO:

trans_list ::= **(** trans_set **)** { **, (** trans_set **)** }

FROM:

trans_item ::= { range_list } | value_range

TO:

trans_item ::= range_list | value_range

As shown below:

```
bins_or_options ::=                                                    // from Annex A.2.11
        ...
      | [ wildcard ] bins_keyword bin_identifier [ [ [ expression ] ] ] = { range_list } [ iff ( expression ) ]
      | [ wildcard] bins_keyword bin_identifier [ [ ] ] = trans_list  [ iff ( expression ) ]
        ...

bins_keyword::= bins | illegal_bins | ignore_bins

range_list ::= value_range { , value_range }

trans_list ::= ( trans_set ) { , ( trans_set ) }

trans_set ::= trans_range_list => trans_range_list { => trans_range_list }

trans_range_list ::=
        trans_item
      | trans_item [ [ * repeat_range ] ]
      | trans_item [ [ *-> repeat_range ] ]
      | trans_item [ [ *= repeat_range ] ]

trans_item ::= range_list  | value_range

repeat_range ::=
        expression
      | expression : expression
```

Syntax 20-3—Transition bin syntax (excerpt from Annex A)

In Section 20.4.1 (top of page 315) remove the extra curly braces, change as shown

`+range_list1+ => +range_list2+`

This specification expands to transitions between each value in `range_list1` and each value in `range_list2`. For example,

`+1,5+ => +6, 7+`

To Yield:

`range_list1 => range_list2`

This specification expands to transitions between each value in `range_list1` and each value in `range_list2`. For example,

`1,5 => 6, 7`

In Section 20.4.1 change the example at the bottom of page 315 (and 316) as shown:

```
bit [4:1] v_a;

covergroup cg @(posedge clk);

    coverpoint v_a
    {
        bins sa = (4 => 5 => 6), (+[7:9],10+=>+11,12+);
        bins sb[] = (4 => 5 => 6), (+[7:9],10+=>+11,12+);
    }
endgroup
```

To Yield:

```
bit [4:1] v_a;
```

```
covergroup cg @(posedge clk);

    coverpoint v_a
    {
        bins sa = (4 => 5 => 6), ([7:9],10 => 11,12);
        bins sb[] = (4 => 5 => 6), ([7:9],10 => 11,12);
    }
endgroup
```

In Section 20.4.3 (top of page 317) change the sentence as shown:

From:
The count of transition bin T0_3 is incremented for transitions (as if by {0,1}->{2,3}):
To:
The count of transition bin T0_3 is incremented for the following transitions (as if by (0,1 -> 2,3)):
